Implementation of Machine Learning to Improve the Decision-making Process of End-of-Usage Products in a Circular Economy

by

Michael Carl Diem

Thesis presented in fulfilment of the requirements for the degree of
Master of Engineering (Engineering Management)
in the Faculty of Engineering at Stellenbosch University

This thesis has also been presented at Reutlingen University, Germany,
in terms of a double-degree agreement.

Supervisor: Prof. Dr. L. Louw

Co-supervisor: Prof. Dr. A. Braun

March 2020

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2020

Abstract

Rising consumption due to growing world population and increasing prosperity, combined with a linear economic system have led to a sharp increase in garbage production, general pollution of the environment and the threat of resource scarcity. At the same time, the perception of environmental protection becomes evident. The Circular Economy (CE) could reduce waste production and decouple economic growth from resource consumption, but most of the products currently in use are not designed for the recovery options of the CE. In addition, the decision-making process regarding following the steps of End-of-Usage (EoU) products has further weaknesses in terms of economic attractiveness for the participants, which leads to low return rates.

This work proposes a model of the decision-making process for laptops, which is divided into two parts. In the first part, the condition of the product on component level is determined by the use of Machine Learning (ML). For this purpose stress factors are developed, which have an impact on the condition of the product. Furthermore, ways are elaborated to capture them, as the product is not physically present. A ML method is selected to process this information. A suitable software application is selected on the basis of defined criteria. In the second part, an economic and ecological evaluation is conducted based on the conditions delivered by the ML process. A possible purchase price is determined on the basis of the costs incurred and the expected selling price. In addition, the emissions saved as a result of the recovery are calculated. In order to demonstrate the potentials of the developed processes and thus validate them, comprehensive data is simulated and a prototype developed. The data is used to train the Artificial Neural Networks (ANNs) and as test cases. This work will contribute to carrying out more advanced decision-making and thereby increase the attractiveness, which should lead to higher return rates of EoU products.

Keywords

Artificial Neural Network; Circular Economy; Classification; Decision-making; Machine Learning; Sustainability

Opsomming

Stygende verbruik as gevolg van die groeiende wêreld bevolking en toenemende welvaart, gekombineerd met 'n lineêre ekonomiese stelsel, het gelei tot 'n skerp toename in vullis produksie, algemene omgewingsbesoedeling en die bedreiging van skaarsheid in hulpbronne. Terselfdertyd word die persepsie van omgewings beskerming uitgelug. Die “Circular Economy” (CE) kan afval produksie verminder en ekonomiese groei van hulpbron verbruik ontkoppel, maar die meeste produkte wat tans in gebruik is, is nie ontwerp vir die herstel opsies van die CE nie. Daarbenewens het die besluitnemingsproses rakende die stappe van “End-of-Usage” (EoU) produkte verdere swakhede in terme van ekonomiese aantreklikheid vir die deelnemers, wat tot lae opbrengskoerse lei.

Hierdie navorsing is in twee verdeel en stel 'n model voor van die besluitnemingsproses. In die eerste deel word die toestand van die produk op komponent vlak bepaal deur die gebruik van Masjienleer (ML). Daarom word stresfaktore ontwikkel wat 'n invloed het op die toestand van die produk. Verder word maniere uitgewerk om dit vas te lê, aangesien die produk nie fisies aanwesig is nie. 'n ML-metode is die geselekteerde metode om hierdie inligting te verwerk. 'n Gepaste sagteware toepassing word op grond van gedefinieerde kriteria geselekteer. In die tweede deel word 'n ekonomiese en ekologiese evaluering gedoen op grond van die toestande wat deur die ML-proses gelewer word. 'n Moontlike koopprys word bepaal op grond van die koste en die verwagte verkoopprys. Daarbenewens word die emissies wat as gevolg van die herstel bespaar is, bereken. Om die potensiaal van die ontwikkelde prosesse te demonstreer en sodoende te valideer, word uitgebreide data gesimuleer en 'n prototipe ontwikkel. Die data word gebruik om die “Artificial Neural Networks” (ANNs) sowel as die toetsgevalle op te lei. Hierdie werk sal bydra tot meer gevorderde besluitneming en sodoende die aantreklikheid verhoog, wat tot hoër opbrengskoerse van EoU-produkte behoort te lei.

Kernwoorde: Kunsmatige neurale netwerk, “Circular Economy”, Klassifiseering, Besluitneming, Masjienleer, Volhoubaarheid.

Acknowledgements

At this point, I would like to thank everyone who supported me during the writing of my master's thesis.

First, I would like to thank Prof. Dr. Anja Braun and Prof. Dr. Louis Louw who supervised my thesis and provided me with professional support. I would also like to thank Prof. Dr. Vera Hummel and Prof. Dr. Daniel Palm, the deans of the study programme 'Digital Industrial Management and Engineering', who made all this possible.

I would like to express my gratitude to my family, who made my studies possible and were always there, ready to support me.

Thank you.

Michael Diem

November 2019

Table of Contents

List of figures

List of tables

List of abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area under the Curve
CE	Circular Economy
CO_2	Carbon Dioxide
CPU	Central Processing Unit
CSV	Comma Separated Value
EoU	End-of-Usage
FPR	False Positive Rate
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ML	Machine Learning
PCF	Product Carbon Footprint
REST	Representational State Transfer
ROC	Receiver Operating Characteristics
SC	Supply Chain
TPR	True Positive Rate
UML	Unified Modeling Language

Chapter 1
Introduction

This chapter serves to introduce the topic of this research work. It describes the background and rationale. Furthermore, it states the problem of the work, which leads to the research question. Subsequently, objectives for the work are defined as well as the research design and methodology. Finally, the thesis outline is presented.

1.1 Background and rationale of the research

The world generates 2.01 billion tonnes of municipal solid waste annually, with at least 33 percent of that [...] not managed in an environmentally safe manner. Worldwide, waste generated per person per day averages 0.74 kilogram (p. 3).

Kaza et al. defined these values in their publication in 2018. Furthermore, they expect global waste to grow to 3.40 billion tonnes by 2050. Reasons for this enormous amount of waste are the increasing production and consumption of goods, caused by a growing population and increasing wealth, and the low percentage of reused resources (Angelis, 2018, p. 19; Ellen MacArthur Foundation, 2018, p. 5). Consequently, the produced waste pollutes the environment and accelerates climate change. One can differentiate waste into two groups, separated by the event of production (Ellen MacArthur Foundation, 2013, p. 15). First, waste produced as a by-product during the production of goods that is not physically incorporated into the product itself. The second group is the end-of-life waste that arises at the end of the functional life of a product. A way to reduce both waste streams is the reuse of products or components and recycling of resources. The quantity of resources needed to produce products has increased significantly of late and is mainly covered by newly extracted resources (Alves Dias et al., 2018).

Figure 1-1 illustrates the development of the extraction of different resources worldwide. It shows clearly the significant increase during the last 40 years. In particular, the volume of electronic waste has risen sharply in recent years and high values of resources are being lost

(Ellen MacArthur Foundation, 2019, p. 6). Some of these resources are finite raw materials, which will lead to a higher demand than supply in the future, if these finite resources are not recovered (Ellen MacArthur Foundation, 2013, p. 14). Electronic products in particular offer great potential (Baldé, Forti, Gray, Kuehr, & Stegmann, 2017). During the last century, the focus has been on the production of goods from virgin materials (newly extracted and non-recycled materials) (Östlin, 2008, p. 3). Major improvements have been made in increasing resource efficiency instead of systematically designing out material leakage and disposal (Ellen MacArthur Foundation, 2013, p. 6). But the awareness of society is shifting to environmental problems and the growing lack of resources (Östlin, 2008, p. 3).

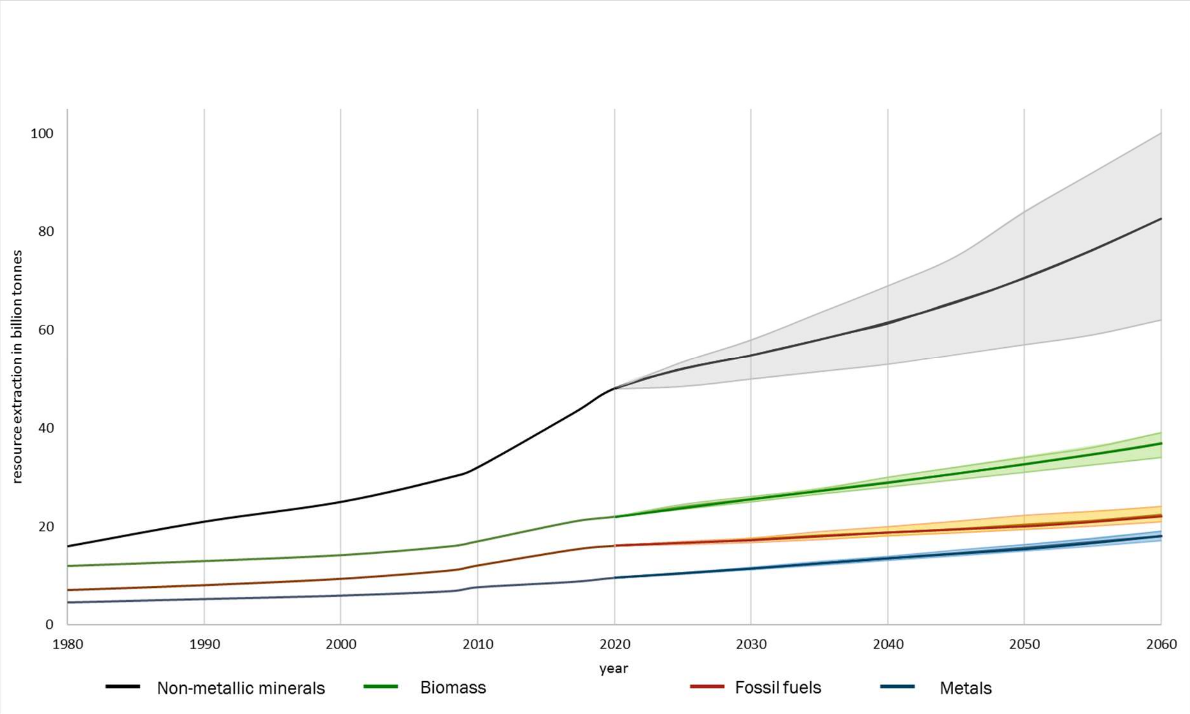


Figure 1-1: Global resource extraction in billion tonnes. Own illustration based on OECD (2019, p. 23), International Resource Panel (2017, p. 29)

Global growth and rising prosperity can be managed only with increased sustainability (Braun, Mandel, & Bauernhansl, 2013, p. 12). An opportunity to reduce waste production and to recover and reuse resources is the so called CE. It tries to replace the 'end-of-life' concept of the linear economy (Ellen MacArthur Foundation, 2013, p. 7). The CE tries to maintain the value of products beyond their initial usage phase (Ellen MacArthur Foundation, 2015b; Sheehan, Braun, Kuhlmann, & Sauer, 2016, p. 455). Furthermore, the Ellen MacArthur Foundation, a current forerunner in research on the CE, states that in the CE "today's goods are tomorrow's resources, forming a virtuous cycle that fosters prosperity in a world of finite resources" (Ellen MacArthur Foundation, 2013, p. 2). Figure 1-2 illustrates the structure of a

CE. It comprises an advanced form of the common supply chain (SC). The difference is the development from a linear to a closed-loop system. The process of the CE starts with a supplier or manufacturer who uses raw material to produce a certain product. This product can be used as a component in another product. After possible substations, the finished good is sold in a workshop to the customer. The customer uses the product until it comes to the end of the time of usage. An EoU product is a product that can be inoperative or operative, but the customer has decided not to use it anymore. The next step is to make the decision whether to remove it or to lead the EoU product back to the CE. If the customer wants to lead the product back, the following steps have to be defined. The possible steps, next to the least preferred removal of the product, are the direct re-use, repair, remanufacture or recycling of the end-of-usage product (Ellen MacArthur Foundation, 2013, p. 14). The black bar represents the decision-making processes of the following steps. The different forms will be elucidated in Chapter 2. Depending on the form, the EoU product enters the SC again, different processes are carried out and the components or the whole product enters the SC at different points.

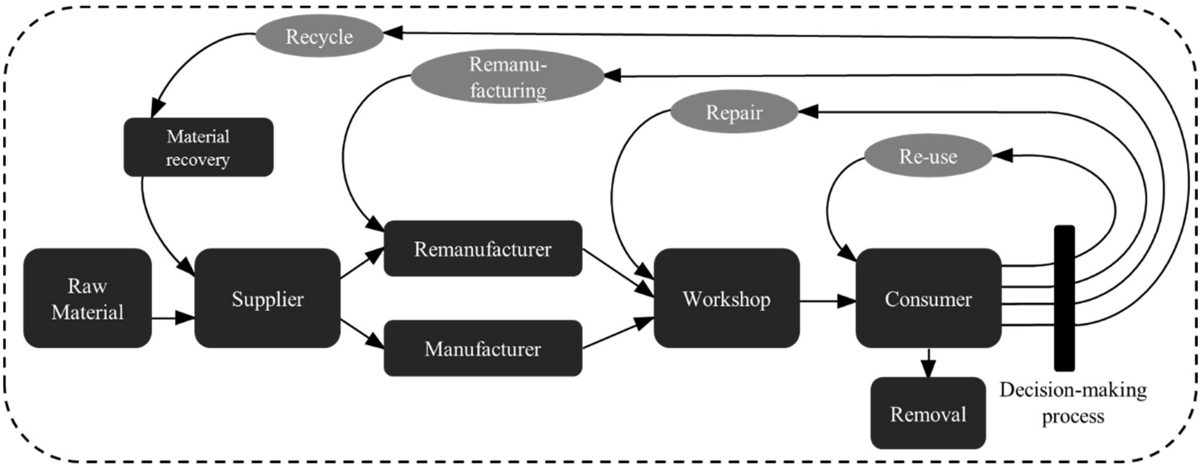


Figure 1-2: System of the circular economy. Own illustration based on Ellen MacArthur Foundation, 2013, p. 24; Gurita, Fröhling, & Bongaerts, 2018, p. 5

The decision-making process has a big influence on the successful implementation of a CE and it is the target of examination in this study. The decision-making process uses information from three different categories, which are economic, environmental and social information (Dunmade, 2004; Goodall, 2014, p. 13). These factors will be described in Section 2.2.3. The current situation and the problems facing the decision-making process concerning the steps for an end-of-usage product will be named in Section 1.2 and will be elucidated in the literature review in Section 2.1.5.

The decision-making process should be improved with regard to uncertainty of profitable reduction to the supply chain, high costs and central implementation (Dyckhoff, Souren, & Keilen, 2004, p. 28; Ellen MacArthur Foundation, 2013, p. 72; Inderfurth, 2004, p. 93; Tilton, 2013, p. 16). An approach to achieve this is the use of ML. Richert and Coelho (2013) define the function of ML in a short and easy form as follows: "Machine Learning teaches machines, how to carry out tasks by themselves" (p. 7). Therefore, the machine learns from provided examples and detects the patterns in the training data. The resulting rule (also called model) of the training process is able to be applied on new data (Richert & Coelho, 2013, p. 8). ML needs much data, but it can provide insights in patterns of data, create forecasts or categorize data. A successful implementation of ML can lead to competitive advantages. Figure 1-3 shows in the Gartner Hype-Cycle that Deep Learning (a form of ML) is a current topic and is the focus of many companies. It is at the peak of inflated expectations, so the public has a big trust in the opportunities and the associated benefits. What is special about ML is that it has been at the peak since 2015, although it was first on the graph in 2014. This shows the importance of ML and that it is here to stay. One reason for the long stay at the peak is that the expectations of ML grow with increasing hardware possibilities. The bigger the processing power and data storage of computers, the bigger the potentials of ML. It also enabled ML to get the attention of practical applications (Pinegger, 2018, p. 16).

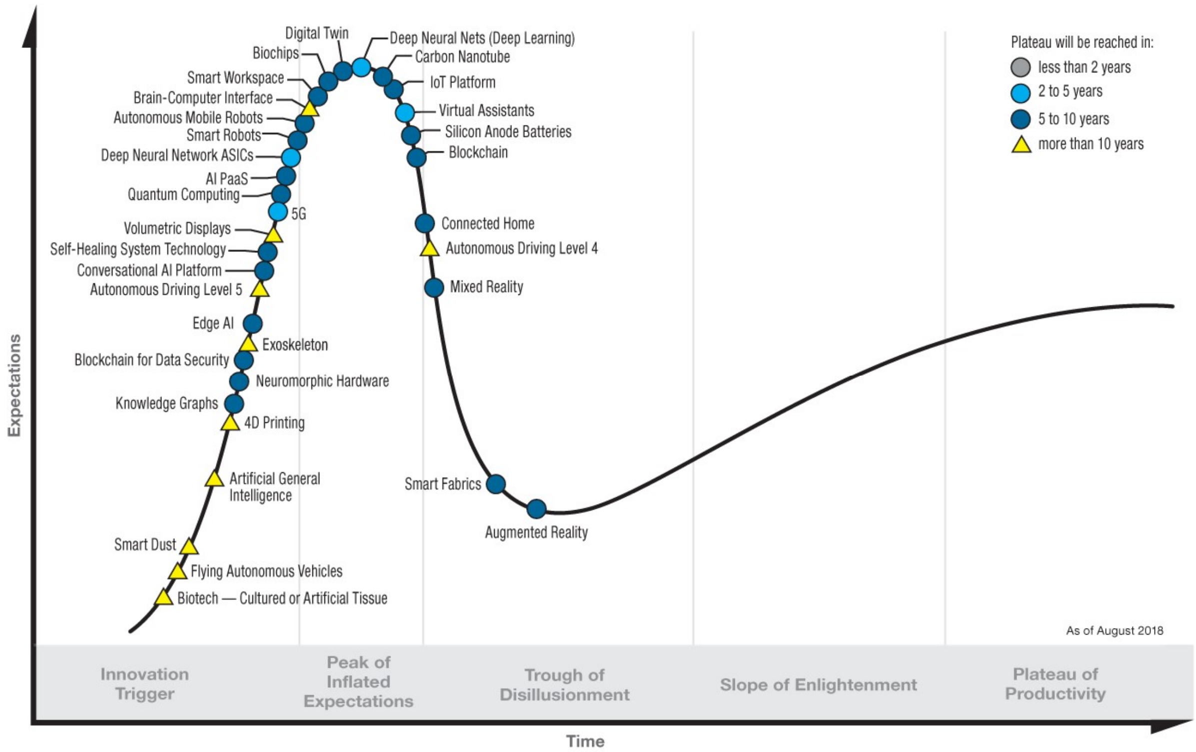


Figure 1-3: Hype Cycle for emerging technologies, 2018 (Gartner, 2018)

1.2 Research problem statement and questions

The problem which builds the basis for this research work, focuses on the decision-making process for EoU products in the CE. The main problem is the low return rate of EoU products back to the SC. The cause of the main problem can be described by the following five sub-problems:

- The classification process of the condition of the EoU product is costly and needs know-how. At the moment it is conducted manually, which makes it expensive and thereby less attractive for customers and companies (Ellen MacArthur Foundation, 2013, p. 72; Tilton, 2013, p. 16)
- The return of EoU products needs a large number of collection institutions and classification instruments. This makes the process expensive. At the moment the main central collection stations are run by respective manufacturers (Dyckhoff et al., 2004, p. 28; Ellen MacArthur Foundation, 2013, p. 80).
- There is high uncertainty of profitable reduction to the SC. This causes low return rates, as often only functional products are returned and companies do not invest in these systems and infrastructure due to the economic uncertainty (Barquet, Rozenfeld, & Forcellini, 2013, p. 1; Goodall, Rosamond, & Harding, 2014, p. 12)
- The major number of products currently in use are not designed for the CE. They are developed for the linear model and are difficult to implement into a reverse supply chain (Hannon, Magnin-Mallez, & Vanthournout, 2016b, p. 3).
- Often there are no incentives for the return of the product (reward system), which makes it less attractive for customers to return the EoU product (Dyckhoff et al., 2004, p. 20).

The research questions which this research work is going to answer are derived from this problem statement. The primary research question is the following:

"How can the decision-making process of EoU products/laptops be improved in order to increase the return rates?"

To answer the primary research question, five secondary research questions must first be answered. They provide the knowledge and findings that serve and clarify the primary research question. Table 1-1 illustrates the secondary research questions and the order in which they should be answered.

Table 1-1: Secondary research questions

Secondary research question 1	What are the problems related to and causes of the low return rates in the CE with regard to the decision-making process?	Section 2.1.5 & Section 3.1
Secondary research question 2	What are the requirements for an improvement of the decision-making process regarding the functions and information required?	Section 3.2
Secondary research question 3	How can the condition of components of laptops not designed for the CE and not physically present be determined?	Section 3.3, 3.4 & 3.5
Secondary research question 4	Which ML method is suitable for determining the condition of the laptop components based on the defined information and which software can best implement this?	Section 3.5 & Section 3.6
Secondary research question 5	How can an EoU laptop recovery be economically and ecologically evaluated?	Section 3.7 & Section 3.8

Answering the first secondary research question identifies the problems of the decision-making process, which leads to the low return rates. Based on the findings gained by answering the first secondary research question, the requirements for an improved decision-making process are derived and defined, which form the answer to the second research question. The third secondary research question serves to develop a process that determines the condition of laptop components without them being physically available. This describes the first part of the decision-making process. For this purpose, it must be answered which information can be used. Once the information has been defined, a method must be selected that can process this information. In addition, software must be selected that can support this method. These selections form the answer to the fourth secondary research question. Finally, for the second part of the decision-making process, an economic and ecological evaluation must be developed that processes the information provided by the first part.

1.3 Research objectives and contribution

The main objective of this work is to improve the decision-making process of a used laptop with regard to the defined requirements. It should be possible to recommend the next step for returning the EoU laptop to the SC in an easy, fast and cost-saving way. In particular, the previously identified problems should be addressed, so that it becomes more attractive for customers and companies to return products to the SC. In the following discussion, further objectives will be pointed out which serve to achieve the main objective and to demonstrate the benefits of the improved decision-making process.

1. A comprehensive understanding of the environment of the CE must be developed. Therefore, participants, relationships and functions should be analysed. In addition, the current problems, as well as the possibilities of the CE must be worked out.
2. The focus of this research will be on the decision-making process. It will thus be necessary to identify the information on which decisions are based. A model of the decision-making process will be developed, which illustrates the scope and functions. The state of the art must be worked out.
3. In the next step, factors must be identified that have a significant influence on the condition of a laptop and how its current condition can be adequately described. This information serves as input and output for the classification process. For this, an understanding of ML must be created with a special focus on classification.
4. Based on the defined information, a ML process is to be defined, which processes this information and determines the condition of the laptop at component level. Thereby criteria should be defined to select suitable software for the implementation of the selected ML method. Furthermore, an adapted evaluation process is to be developed, which is based on the provided component conditions of the laptop.
5. The theoretically developed processes are to be implemented in a prototype. For this purpose, comprehensive data must be simulated to train the ML method and test the developed prototype.
6. The developed decision-making process is to be verified and validated. This is performed on the basis of the developed prototype and surveys of experts and potential users of the process.

This research contributes to research into the application of ML in the CE. The decision-making process regarding the further stages of an EoU product are to be improved. On the one hand, customers have a benefit, because they have the opportunity to sell their used product or to buy a used and thus cheaper product. On the other hand, companies benefit, because they can more easily coordinate their after-sales activities and get cheap raw materials or spare parts, which in turn also benefits the customer (Rossé, Stuchtey, & Vanthournout, 2016, p. 7). In addition, it facilitates compliance with already existing or future regulations regarding the withdrawal of EoU products.

Another important aspect is the protection of the environment. It saves resources and reduces waste and emissions, as a refurbished or compounded product uses less energy than a completely newly produced one (Angelis, 2018, p. 12; Pineyro & Viera, 2018, p. 2).

The structure and procedure of the developed decision-making process should serve as a basis for further research projects. The implementation and testing on the basis of simulated data is intended to uncover the potentials of the process and encourage practical implementation.

1.4 Research design and methodology

Literature names three different types of research purposes, which influence the research question and thereby the research design and the used methodology. They are called exploratory, descriptive and explanatory studies (Saunders, Lewis, & Thornhill, 2009, p. 138). This study is not clearly assignable to one of these three types. It is a mixture of descriptive and explanatory and is therefore to be called a descripto-explanatory study. On the one hand, it describes and forms certain situations and processes in the environment of CE and ML. On the other hand, it shows relationships between participants and variables of the system as well.

The research philosophy relates to the development of knowledge and the nature of knowledge (Saunders et al., 2009, p. 107). It contains important assumptions, which will underpin the research strategy and the methods which will be conducted (Saunders et al., 2009, p. 108). Saunders et al. (2009, p. 108) differentiate between: (i) positivism, (ii) realism, (iii) interpretivism and (iv) pragmatism. This research study adopted the pragmatism research philosophy. The reason is that the study focuses on practically applied research, integrating different views to answer the research question. The other research philosophies are not discussed here, because they are out of the scope of this research study.

Furthermore, the study uses a mixed-methods design, because it utilises quantitative and qualitative methods. A qualitative research design is used to gain and explore knowledge on the CE environment, decision-making and ML methods. For this purpose, a literature review is carried out. The elaborated knowledge is used to develop a prototype. Testing the prototype resembles an experiment, which describes a qualitative research design. The prototype also serves as an illustration. It is used to conduct surveys to determine the implementation of the defined requirements. When carrying out the literature review, care is taken to ensure that the information comes from reputable and up-to-date sources. When determining the state-of-the-art the sources should not be older than ten years. For basic definition and theories, older sources can be used, as the original reference is given. When simulating the data, the assumptions made are based on research, but do not reflect reality.

This study follows a deductive research approach. The deductive approach describes the process whereby a theory is elaborated on and a fitting research strategy selected in order to verify this theory. In contrast to this is the inductive approach, in which data is collected and analysed to develop a theory (Saunders et al., 2009, p. 124). In this research work, the theory suggests that the use of machine learning is able to improve the decision-making process of EoU products in the CE. After implementing the developed process, the results will be analysed in order to verify or deny the elaborated solution.

Due to the fact that this work will develop a prototype in the form of a software program, this work partly follows a software development methodology. For this purpose, the waterfall model was chosen, developed by Royce (1970). Figure 1-4 illustrates the steps to develop the software program. The process is not a one-way as the original waterfall model. Interactions between the individual steps are possible.

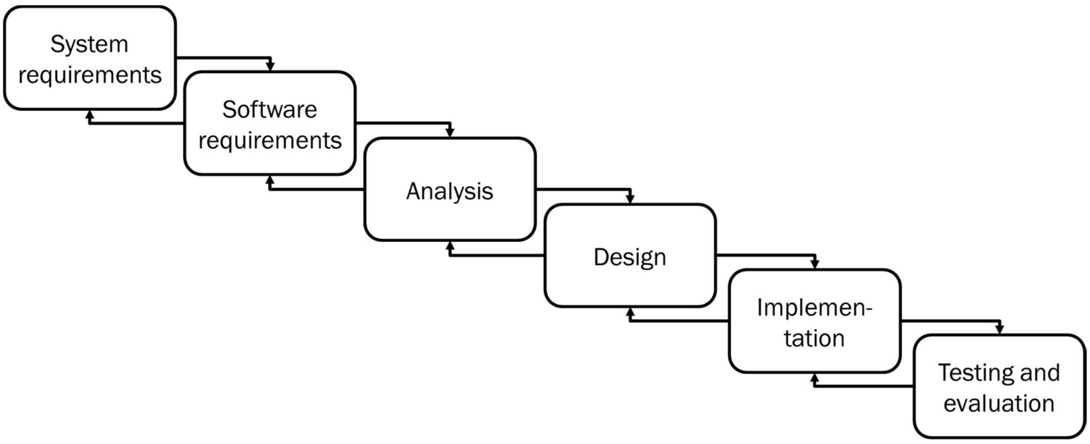


Figure 1-4: The waterfall methodology of this research work. Own illustration based on Goodall (2014, p. 3); Royce (1970, p. 330) The stages of the waterfall method of this work are the requirements of the system, requirements of the software, analysis, design, implementation and the evaluation and testing.

1. Requirements of the system: In this stage the functions and parameters of the decision-making process are identified, which will be implemented in the software program. This is done on the basis of current problems and weaknesses. For this, the decision-making process is divided in parts and analysed.
2. Requirements of the software: This section defines the required functionalities and features of the software. For this purpose, inter alia, the findings from the analysis of the decision-making process are used. Furthermore, the criteria for the software selection are defined.
3. Analysis: The analysis phase evaluates possible software solutions by conducting a pairwise comparison relative to the criteria defined in the previous stage. In addition, a ML method suitable for the classification process is selected.
4. Design: Within the design phase the structure of the selected ML method is developed. Furthermore, a class diagram and a flow chart are created for the economic and ecological evaluation, which serve as the basis of the implementation.
5. Implementation: The information from the previous steps is used for coding and implementing the decision-making process. The interface between the individual system parts is established.
6. The created software system is tested relative to the defined requirements. In addition, two surveys are conducted in order to determine the implementation of the defined requirements. Therefore, experts as well as potential users of the developed process will be asked for their opinion.

1.5 Thesis outline

The structure of the thesis is derived from the defined research questions, objectives and the previously presented waterfall model. It can be divided into five parts. The first part includes the first chapter. Information on the entire thesis is presented here. The second part describes the literature review. In this part, only information will be researched and reproduced. No own contents will be produced yet. In the third part the own development of content begins. Here, the decision-making process is first developed theoretically using the previously researched information. The fourth part comprises the Chapters 4 - 6. The theoretically developed process is implemented. Nevertheless, information on individual topics will continue to be researched. In this part, the created contents are also verified and validated. The fifth part follows at the end. No concrete contents are created here any more. However, conclusions are still being drawn. In addition, the thesis is summarised, critically reflected on and further research tasks are proposed.

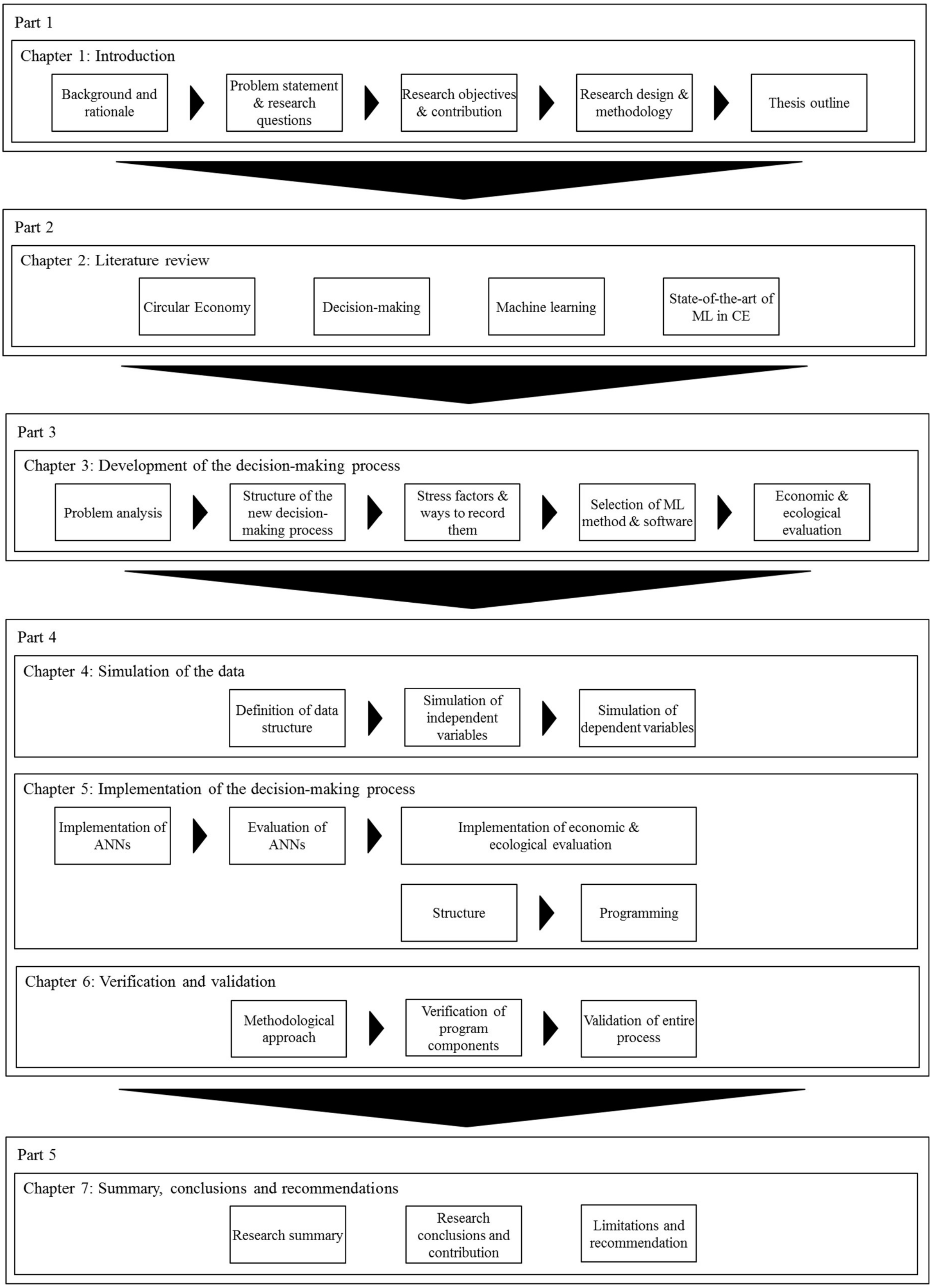


Figure 1-5: Structure of the thesis

1.6 Chapter summary

This chapter introduces the research project and gives background information on the subject of CE and ML, as well as the rationale of the research. Furthermore, the problem to be treated is stated and the research question derived. The defined objectives and research contribution, as well as the applied research methodology and design are shown. The chapter concludes with an outline of the thesis.

Chapter 2
Literature review

This chapter provides an overview of relevant literature that has been analysed in order to build the basis of this research study. The first part deals with the CE, its structure, possibilities and problems. The second part describes a review on the decision-making process in general and of EoU products, since this research work focuses on this part of the CE. The final part analyses the subject of ML and the state of the art.

2.1 Circular economy

This part of the chapter deals with the theory of CE. For this purpose, the first part develops a definition of CE which is valid for this research work. Subsequently, the work discusses and explains the structure, principles and processes within the CE. The next part elaborates the possibilities and advantages of the CE and identifies the current problems of the CE.

2.1.1 Conceptual distinction

The first time a critical view of the traditional linear economic system led to the use of the term 'Circular Economy' was in 1990 by Pearce and Turner (Rizos, Tuokko, & Behrens, 2017, p. 2). Based on the principle that everything is an input to everything else, Pearce and Turner (1990) developed a model of the CE. Rizos et al. mentioned in their review of 2017, that there are often different interpretations of the concept of CE. That is the reason this study first develops a definition and understanding valid for this work. Referring to Rizos et al., two ways to define CE exist. The first one provides resource-oriented definitions, which focus closed loops of material flows and reduction of consumption of virgin resources and environmental impacts. The second section moves beyond the consumption of resources and deals with forms of energy used and economic and social effects (Rizos et al. 2017).

Sauvé et al. (2016, p. 53) states that CE "aims to decouple prosperity from resource consumption, i.e., how can we consume goods and services and yet not depend on extraction of virgin resources thus ensure closed loops that will prevent the eventual disposal of consumed goods in landfill sites." With this statement, they focus on the closed loop structure to prevent disposal. In addition, they also address the potential benefits of reusing resources by claiming "[CE] proposes a system where reuse and recycling provide substitutes to the use of raw virgin materials. By reducing our dependency on such resources, it improves our ability […] to meet their needs. The circular economy makes sustainability more likely" (p. 53). The closed-loop structure can lead to reduced dependencies with regard to resource supply. Furthermore, they describe the CE as a factor for a sustainable economy.

Another approach to define CE with focus on the resources is the one by Preston (2012). He describes CE as "an approach that would transform the function of resources in the economy. Waste from factories would become a valuable input to another process – and products could be repaired, reused or upgraded instead of thrown away" (p. 1). This approach also highlights the transition from the linear to the circular system, which they describe as the transition from waste to input. Furthermore, they mention possible recovery options of the CE.

The European Environment Agency (2014) emphasises the use of waste as a resource for the value chain as well. It claims that CE "refers mainly to physical and material resource aspects of the economy […]. It focuses on recycling, limiting and reusing the physical inputs to the economy, and using waste as a resource, leading to reduced primary resource consumption" (p. 11). The possible positive effect of reducing resource consumption is also used here in the description.

The resource usage reduction also plays a role in the view of Heck (2006). However, he goes one step further and takes up the energy question. He directs one's attention, besides the waste and resource problem, also to the "sustainable availability of energy" (p. 6). Bastein et al. also embrace the transition to renewable and sustainable energy supply. Furthermore, they speak of a change in our mindsets, in which waste cannot simply be passed on to nature but must be avoided already in the design phase of products and systems. Moreover Bastein et al. see CE as an opportunity to increase competitiveness and generate employment.

One organisation, which deals a lot with the topic of CE is the Ellen MacArthur Foundation, and has written several meaningful and often quoted reports. The Ellen MacArthur Foundation (2013) gives the following definition of CE:

> A circular economy is an industrial system that is restorative or regenerative by intention and design. It replaces the 'end-of-life' concept with restoration, shifts towards the use of renewable energy, eliminates the use of toxic chemicals, which impair reuse, and aims for the elimination of waste through the superior design of materials, products, systems, and, within this, business models (p. 7).

This definition is a good summary of the definition approaches listed above. It contains the important aspects and is easy to understand. In addition, it stresses the replacement of the end-of-life concept with a closed-loop model, which is the focus of this research work. Therefore, this is the valid definition for the term 'Circular Economy' for the further course of this research work. An important reason for this decision is that it is one of the most frequently used definitions.

2.1.2 Structure of the circular economy

The cradle to cradle design concept developed by Braungart and McDonough considers materials as nutrients and describes a framework of an industrial system. They separate the nutrients into biological and technical nutrients. The concept does not focus on reducing or eliminating negative environmental effects. Rather, it focuses on the positive effects of material flows by increasing the effectiveness and quality of resource use (Braungart, McDonough, & Bollinger, 2006, p. 1343).

The Ellen MacArthur Foundation has introduced a structure of the CE system. Like the cradle to cradle concept, this consists of the cycles of the biological and the technical nutrients. For this research work, only the cycle of technical nutrients is relevant. Figure 2-1 shows the technical cycle of the CE.

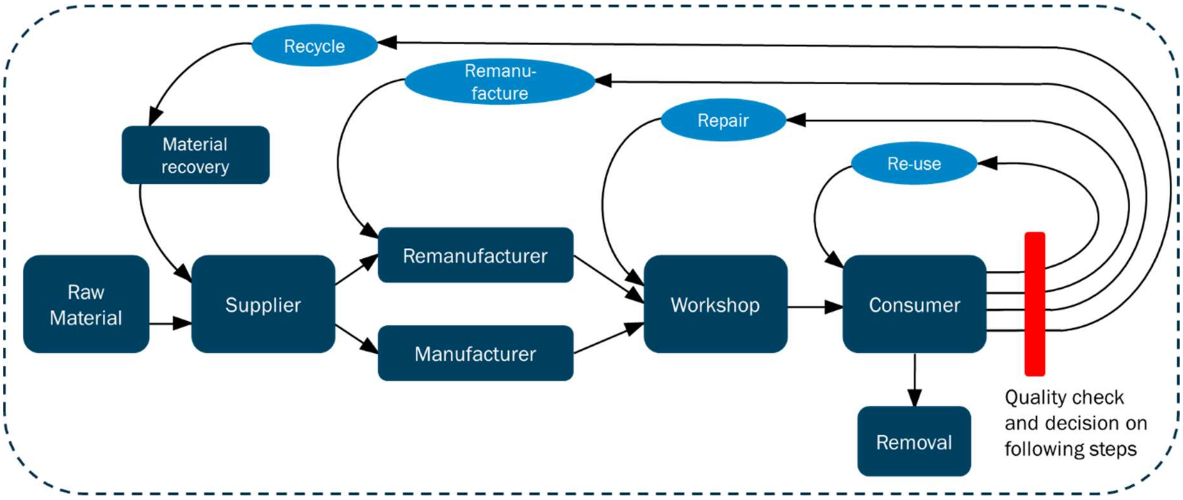


Figure 2-1: System of the circular economy. Own illustration based on Ellen MacArthur Foundation, 2013, p. 24

Before resources can circulate in the CE, they must enter the cycle as raw materials. There, they are either used by a supplier to manufacture a component or are used directly in a marketable product. The manufacturer sells the produced product to the customer via a workshop. The customer uses the product until it achieves either the end-of-life or end-of-usage point. The end-of-life point is reached when the product is no longer able to perform the functions intended for it (Friege, 2012, p. 6; Jun, Cusin, Kiritsis, & Xirouchakis, 2007).The end-of-usage point occurs when the customer decides to no longer use the product (Mukherjee, Mondal, & Chakraborty, 2017, p. 78). The end-of-usage point can be the same as the end-of-life point. However, this point may occur when the customer decides to buy a successor or substitute product or the product is no longer needed. At this point of the CE there are two possible ways for the EoU product. It is apparently easier for the customer to dispose of the EoU product (Dyckhoff et al., 2004). It then becomes waste, because it leaves the CE. It can be used as energy recovery or ends up in a landfill (Östlin, 2008, pp. 40–41). The principals of the CE, which the following section discusses in detail, try to avoid that way of removal. The other possibility is that the customer decides to keep the product in the CE by returning the product to the SC. This requires the product to first undergo a quality check, which is represented as a red bar in Figure 2-1. On the basis of the results of the quality check, the following steps are determined. The EoU product goes to one of the following product recovery options, depending on the condition of the product and external circumstances.

1. Recycling: The functionality of the product or component is lost. The purpose of recycling is to reuse the materials of the product (Thierry, Salomon, van Nunen, & van

Wassenhove, 1995, p. 120). Three different types of recycling exist (Ellen MacArthur Foundation, 2013, p. 25).

   a. Functional recycling: Recovery of materials for the same purpose or other purpose which requires the same quality and functionality.
   b. Downcycling: Recovery of materials, transforming them into materials of lower quality and functionality.
   c. Upcycling: Recovery of materials, transforming them into materials of higher quality and functionality.

2. Remanufacturing: The product is disassembled and the defective components are replaced with new or as good as new components. In addition, the product is generally reconditioned so that it achieves an as good as new condition and the customer gets a guarantee on the entire product (Agrawal, Atasu, & van Ittersum, 2015, p. 1; Goodall, 2014, p. 10). Remanufacturing can even include technological upgrading (Hübner, Himpelmann, Melnitzky, Stahel, & Hübner, 2006, p. 29; Thierry et al., 1995, p. 119). This includes functions that the original product did not have, but were requested by the customer at the time. These products are offered by the remanufacturer itself or are sold to a third-party broker (Ferguson, Fleischmann, & Souza, 2011, p. 774).
3. Repair: Defective components in the product are replaced in order to reinstate the functionalities. Only these components regain an as good as new condition and are therefore guaranteed (Amezquita, Hammond, Salazar, & Bras, 1995, p. 1; Andrew-Munot, N Ibrahim, & Junaidi, 2015, p. 2). A repaired product is generally of less quality than a remanufactured or new product (Thierry et al., 1995, p. 118).
4. Reuse: Product is used for the same purpose for which it was manufactured, without replacing parts, only receiving refurbishing and superficial cleaning. The term also covers the process of removing a functional component from a product and incorporating it into another product. The component is reused and the product is remanufactured (Amezquita et al., 1995, p. 1; Ellen MacArthur Foundation, 2013, p. 25; Paterson, Kao, Ijomah, & Windmill, 2018, p. 6).

Table 2-1 shows a summary of the previous named recovery options. They are examined according to the level of disassembly, quality requirements and resulting product.

Table 2-1: Recovery options in the CE. Own illustration based on (Thierry et al., 1995, p. 120)

	Level of disassembly	**Quality requirements**	**Resulting product**
Reuse	No disassembly	Functional product	Same product
Repair	Product level	Functional product. Guarantee on replaced component	Same product
Remanufacturing	Part level	Functional product. Check of all modules and sometimes upgrade. As good as new condition. Guarantee on whole product	As good as new product (with more functionalities)
Recycling	Material level	Useable for new manufacturing process	Completely new product.

The principles of CE consider the direct reuse as the best recovery option, since all resources are used here and it needs the lowest amount of energy (Amezquita et al., 1995, p. 1). Therefore, this option should be used if the condition of the EoU makes it possible. Figure 2-2 shows the order in which the previously options are preferred.

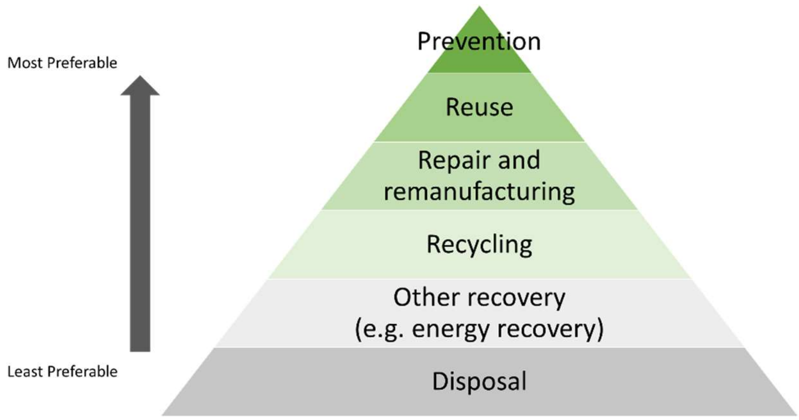


Figure 2-2: Hierarchy of options of end-of-life/ -usage products. Own illustration based on European Parliament and the Council of the European Union, 2008, p. 10; Long et al., 2016, p. 7

This is followed by repair and remanufacturing in the preference list. There is only a very slight difference between these two recovery options, as the product is disassembled for both and new parts are incorporated. In the CE, recycling is the least preferred method, as this requires the most energy and parts of the resource are lost. One of the two possibilities where materials leave the CE is energy recovery, where at least energy is recovered from materials before they leave the CE. With the disposal option, the materials and resources end up directly in a landfill. However, this does not belong to the recovery options. At the top of the pyramid is prevention, which is actually not a recovery option. However, from an ecological and resource-saving point of view, it would be the best to avoid the generation of an EoU product. The options with a green colour as background take place in the CE (reuse, repair/remanufacturing, recycling) or is in the sense of the CE (prevention). The options with a grey background show activities where the resource stops circulating in the CE.

Depending on the recovery option, the product returns to the SC at a different point. If the product should be reused, it is sent directly to a new customer or to a workshop and then sold to a new customer (it is often cleaned beforehand). For the repair, it is sent to a workshop and is repaired there and then sold again. In order to be remanufactured, the product comes to a remanufacturer who may also be a manufacturer. There is a shift in the roles, as the customer becomes the supplier (Tilton, 2013, p. 20). This is a consequence of the previously explained transition from waste to resource. Remanufacturing is mostly performed by third party remanufacturers (Amezquita et al., 1995, p. 3). It goes through the necessary processing steps and is sold via the workshop. Recycled materials are led together with the new raw materials to the supplier or the manufacturer.

2.1.3 Principles

The CE follows several principles. The Ellen MacArthur Foundation (2013) names three main principles:

1. *Design out waste:* There should be no waste in the CE. Therefore, the products and processes are designed for the CE. The design of a product is developed for disassembly, in order to remanufacture and repair products, and reuse them (Ellen MacArthur Foundation, 2013, p. 7).
2. *Differentiation between consumables and durables:* The CE makes a clear distinction between consumables and durables. Consumables are to be produced primarily from biological ingredients. They can enter the biosphere without concern. This can be done directly or through cascading uses (Ellen MacArthur Foundation, 2013, p. 7).
3. *Use of renewable energy:* The energy required in the CE for processes such as remanufacturing and transport should come from renewable sources (Ellen MacArthur Foundation, 2013, p. 7).

The explained principles, on which the theory of CE is based, are extended by approaches to develop the benefits of the CE. These lead to the definition of the recovery options of the CE and to the preference ranking.

1. *Using the inner cycle:* The CE always tries to take the inner circle. This reflects the preference for the recovery options mentioned in Section 2.1.2. The less a product needs to be changed and processed, the less energy, resources and labour will be used (Ellen MacArthur Foundation, 2013, p. 7). This reduces the impact on the environment.
2. *Circling longer:* The product or the resources should be kept as long as possible in the CE. It is possible to extend the duration of a cycle (e.g. to increase the lifetime of the product) or to increase the number of cycles (Ellen MacArthur Foundation, 2013, p. 7).
3. *Cascaded use:* Resources should be reused more often. This is done by reusing them anyway, even if their quality and functionality are no longer sufficient for the original purpose. They are used for other purposes, in the meaning of downcycling, which is described in Section 2.1.2 (Ellen MacArthur Foundation, 2013, p. 7).
4. *Pure circles:* Increasing collection and redistribution efficiency by using uncontaminated material. Less material has to be disposed of (Ellen MacArthur Foundation, 2013, p. 7).

Another principle that the CE pursues describes the customer's transition to a user. Particularly in the case of technical nutrients, business tries to lease, rent or share products to users (Ellen MacArthur Foundation, 2013, p. 7). This should lead to a maximisation of their utilisation during the lifetime of the product (Ellen MacArthur Foundation, 2015a, p. 25).

2.1.4 Possibilities and advantages

The CE is interesting because it can be a more environmentally friendly and sustainable solution than the linear economy. This section shows and explains the advantages and possibilities of the CE compared to the 'take-make-dispose' approach of the linear economy. Amezquita et al. (1995) summarised the driving factors of the CE under the following points: Ecology, legislation and economics. Table 2-2 shows the different advantages of the CE in more detail.

Table 2-2: Advantages of the CE

No.	Advantage	Source
1.	Reduction of waste	Amezquita et al., 1995, p. 3; Angelis, 2018, p. 23; Ellen MacArthur Foundation, 2018, p. 3; Goodall, 2014, p. 1; Paterson et al., 2018, p. 4
2.	Reduction of required energy and produced emissions	Li, Wu, Jin, & Lai, 2017, p. 581; Matsumoto & Ijomah, 2013, p. 390; McKinsey Global Institute, 2011, p. 148; Quariguasi-Frota-Neto & Bloemhof, 2012, p. 112; Sundin, Sakao, Lindahl, Kao, & Joungerious, 2016, p. 27
3.	Reduction of required resources and volatility of markets. Avoidance of resource dependency	Ellen MacArthur Foundation, 2013, p. 10; Goodall, 2014, p. 1; McKinsey Global Institute, 2011, p. 29; Zlamparet et al., 2017, p. 127
4.	Saving of material costs	Ellen MacArthur Foundation, 2013, p. 6; Rossé et al., 2016, p. 7
5.	Entering new customer segments	Ellen MacArthur Foundation, 2013, p. 72; Hannon, Kuhlmann, & Thaidigsmann, 2016a, p. 23; Sundin, Sakao, Lindahl, Kao, & Joungerious, 2016, p. 21
6.	Generation of employment and economic growth	Angelis, 2018, p. 23; Morgan & Mitchell, 2015, p. 20; Rossé et al., 2016, p. 5; Sundin et al., 2016, p. 28

The first advantage of the CE has already been mentioned, describing the reduction of waste. The further circulation within the CE prevents the generation of waste, which is harmful to the environment. This is because fewer products have to be disposed as waste and fewer new products have to be produced due to reuse. In the production process waste is generated as a by-product of packaging material or scrap (Sheehan et al., 2016).

This point belongs to the ecological aspects, as well as to the next advantage: reduction of energy requirements and emissions produced. Fewer manufacturing steps are necessary because components of the product are only manufactured and assembled with existing components to form a product. This means that remanufacturing needs less energy than producing a completely new product. The reasons for this are fewer transport routes and fewer processing steps. Since the generation of energy currently still produces many emissions, less energy required also means fewer emissions produced.

The third advantage, reduction of required resources and volatility of markets and thereby avoidance of resource dependency, is both ecological and economical. The ecological aspect is that in general less resources are needed, which is a more sustainable way of production. This is important because currently about 1.5 up to 2 times more resources are used to make products than can be produced by the world (Ellen MacArthur Foundation, 2015a, p. 17). The CE reduces the volatility of the markets by the fact that demand is linked to the return of products and thus to the return of resources. This avoids resource bottlenecks. If you keep the products in your own cycle, you become to some extent independent from resource suppliers. The McKinsey Global Institute (2011, p. 29) states that the volatility of resource prices is at an all-time high (with the exception of energy in the 1970s).

In addition, the saving of material costs is a big advantage of the CE. It could increase a company's profitability and offers customers a better price for products. There is a great potential here, which is shown by the fact that, according to Statistisches Bundesamt (Destatis) (2018, p. 8), material costs account 41.97 % of gross production costs in Germany. The use of new technology could lead to an increase of resource productivity by up to 3 % (Rossé et al., 2016, p. 5). The Ellen MacArthur Foundation (2013, p. 6) speaks of an opportunity of material cost saving for the EU manufacturing sector of up to USD 380 billion in a transition scenario and up to USD 630 billion in an advanced scenario.

A further advantage of the CE is that it creates the opportunity for companies to develop new customer segments. There are two possible reasons for this. On the one hand, reused products appeal to customers who pay attention to sustainability and environmental protection. On the other hand, used or remanufactured products can be offered for a lower price. This makes it possible for customers with lower purchasing power to buy such a product even though their purchasing power would not be sufficient to buy the new product. This could lead to an increase in sales, as the target group is bigger.

The last advantage listed in Table 2-2 is the generation of jobs and economic growth (Morgan & Mitchell, 2015, p. 20). Since remanufacturing and repair processes consist to a large extent of manual steps, CE will create jobs for both low-skilled and for qualified workers. However, since highly qualified workers must first implement the concept and there is still a great need for research in this field. There some jobs will become available for highly qualified employees too (Rossé et al., 2016, p. 7). Furthermore, the World Economic Forum (2019) assesses the value of electrical waste at approximate USD 62.5 billion annually. This shows the opportunity for economic growth and cost savings.

Another reason, which is not listed Table 2-2 is to focus more on the concept of CE and to implement parts of it are legal regulations and rules. The WEEE (waste electrical and electronic equipment) Directive pursues higher collection rates, the goal of achieving higher reuse, remanufacturing and recycling rates for electrical waste (European Parliament and the Council of the European Union, 2012). The WEEE Directive builds on producer responsibility. There are incentives for producers for supporting these goals by attributing the financial responsibility for collection and recovery of their own products (Gurita et al., 2018, p. 4). Furthermore, the sustainable action of companies plays an increasing role over time and thus has a decisive influence on the company's public image and the relationship to their customers. In summary, CE can lead to competitive advantages for companies in the present and the future.

2.1.5 Problems and challenges

After elaborating the benefits of the CE, the problems of why the CE is not yet widely implemented and why it is not fully benefiting from its potential are being addressed. Table 2-3 lists the main problems and challenges of the CE.

The first challenge describes the fact that resource prices for the production of a completely new product are currently often lower than the reconditioning costs of a used product. This is mainly due to the fact that the original production process is optimized and the reprocessing consists to a large extent of manual work steps, which generates high costs. Especially in Germany, manual labour has a strong impact on the costs.

The next challenge stresses the problem of the development of shorter life cycles (Ferguson et al., 2011, p. 774). In order to survive in the market, companies develop new products for the market at shorter intervals in order to satisfy the customer. In addition, in the course of customer orientation, considerably more variants and different products are offered. As a result, the complexity that CE processes have to handle is increasing.

Table 2-3: Challenges and problems of the CE

No.	Challenge	Source
1.	Low resource prices compared to labour costs. Low automation level	Ellen MacArthur Foundation, 2013, pp. 14–15; Lange, 2017, p. 27
2.	Short lifetime cycles and many different products	Deng et al., 2006, pp. 6945–6946; Goodall et al., 2015, p. 2; Gurita et al., 2018, p. 1; Long et al., 2016, p. 2; Macauley, Palmer, & Shih, 2003, p. 13; Ramadoss, Alam, & Seeram, 2018, p. 56
3.	Uncertainties of return	Andrew-Munot et al., 2015, p. 3; Barquet et al., 2013, p. 1; Ellen MacArthur Foundation, 2018, p. 11; Han, Wu, Yang, & Shang, 2016, p. 63; van Nunen & Zuidwijk, 2004, p. 43
4.	Complex process. Need of know-how with regards to disassembly and return	Goodall et al., 2015, p. 2; Guide Jr., V. Daniel R., 2000, p. 468; Lange, 2017, p. 27; Long et al., 2016, p. 9; Singh & Jain, 2016, p. 2
5.	Infrastructure, system and preselection	Barquet et al., 2013, p. 5; Dyckhoff et al., 2004, p. 28; Ellen MacArthur Foundation, 2013, p. 9; Engel, Stuchtey, & Vanthournout, 2016, p. 12; Friege, Oberdörfer, & Günther, 2015, p. 224; Hannon et al., 2016a, p. 23; Parajuly & Wenzel, 2017, p. 279; Zlamparet et al., 2017, p. 132
6.	Rebound and cannibalisation effect	Agrawal et al., 2015, p. 1; McKinsey Global Institute, 2011, p. 142; Rossé et al., 2016, p. 5; Zink & Geyer, 2017
7.	Products are not designed for CE	Amezquita et al., 1995, p. 2; McKinsey Global Institute, 2011, p. 148; Thierry et al., 1995, p. 114
8.	Missing attractiveness for owner of EoU product. Missing understanding of potentials of CE	Ellen MacArthur Foundation, 2013, p. 9; Ellen MacArthur Foundation, 2015a, p. 21; Ellen MacArthur Foundation, 2018, p. 8; Hannon et al., 2016a, p. 23; Parajuly & Wenzel, 2017, p. 279

The next point deals with one of the biggest problems the CE has. It describes the uncertainties caused by a closed-loop SC, besides the uncertainty regarding demand, which is also present in the linear economy. These make the decision-making about reverse processes more difficult than traditional forward manufacturing processes. Goodall (2014) divides the uncertainties from different literatures into three categories. These are uncertainties regarding (i) condition of the returning product, (ii) design and physical structure, as well as (iii) timing of return and the number of returning products. Point (ii) arises from a missing information flow (Goodall, 2014, p. 25), whereas points (i) and (iii) can only be solved by forecasts, which are always inaccurate (Box, 1976) or require the application of new technologies such as the 'Internet of Things'. These uncertainties also make it difficult to ensure economic return of the products, which means that products are more likely to be disposed of than losing money by returning the product. It also leads to fewer investments when there are too many uncertainties about economic profitability, because profitability is the main driver for engagement (Guide Jr., V. Daniel R., 2000; Inderfurth, 2004, p. 92). The operating network has to be financially attractive for the companies over the long term (Braun, Kleine-Moellhoff, Reichenberger, & Seiter, 2018, p. 2).

The fourth challenge has a strong relationship to the first challenge. In normal forward manufacturing, the employees are usually only responsible for a few products of one brand. In the CE, they have to process all products of a product category. Since the processes are mostly carried out manually, an enormous amount of expertise, which employees must possess, is necessary.

The infrastructure, which the CE requires is another challenge. This includes in particular the collection points to which the EoU products are transported in order to be processed and stored. In addition, reverse logistics networks must be established. This infrastructure should be able to handle different products of different brands.

Furthermore, special effects can occur in the CE. These include the rebound and the cannibalisation effect. The rebound effect describes the fact that the supply of additional resources through recycling reduces the price of these resources, as there are more resources available on the market as a result. This makes the production of new products more attractive, as the main cost drivers are the costs for raw material (Rossé et al., 2016, p. 5). As a result, the focus shifts to the new production and fewer products are reused or refurbished, since it is more profitable to produce than to recover. The cannibalisation effect, on the other hand, describes a marketing problem. If a manufacturer offers used products of his brand, this can have negative effects on the reputation and external appearance of his brand. With a product of this brand, the customer can no longer differentiate so well from customers with less purchasing power, because they can afford this brand. Thus, the prestige of the brand suffers from the offer of reused products. The remanufactured products can also cannibalize the demand for the new products (Agrawal et al., 2015, p. 1).

One challenge that currently arises most frequently in the linear economy is that most products, currently in use are not designed for the CE. As a consequence, it is very difficult to determine their current condition and reprocessing is difficult or not possible at all due to their construction. It is not possible to disassemble those products, because of manufacturing procedures such as gluing, because the product will be destroyed or reprocessing is so difficult that it makes no economic sense (Hannon et al., 2016a, p. 23). This problem has been known for quite some time (which is also reflected in the dates of the referenced literature) but from an economic and legal point of view, manufacturers have not done anything about it so far. Some products have been designed to break after a certain period of time in order to ensure a steady cash flow (Hübner, 2013).

The last challenge in Table 2-3 is that there are rarely incentives for customers to return their product to the SC at the end of its useful life. As long as the customers are not obliged to take the EoU product to a collection point or have some benefit for returning it, they will normally dispose of the product in the easiest way. There are already companies which buy up used products, and the customer thus receives a payment for them, but this is not yet widespread and condition recording is relatively difficult.

2.2 Decision-making process

This section of the chapter analyses the decision-making process in general and specifically that of EoU products in the CE. In addition, the relevant factors for the decision and the state of the art are worked out.

2.2.1 Steps of a general decision-making process

Decision means the conscious selection of an alternative action from at least two available actions (Blume, 2003, p. 83). Blume (2003) distinguishes between two decision-making situations: decision under risk and decision under uncertainty. Deciding under risk means that the decision-maker has clear probability distributions about the occurrence of possible consequences of an alternative action. Deciding under uncertainty describes the situation that the decision-maker is aware of the relevant and possible consequences of an alternative action, but the probability distribution is unknown or incalculable. Therefore, the aim in this case is to collect data and information in order to reduce uncertainty and to decide more under risk. Expertise and experience can also help to reduce uncertainty. (Mentis, 2015, p. 3)

Avramenko and Kraslawski (2008) divide the general decision-making process into seven steps. The decision-making process starts with the definition of the problem to be treated (Guo, 2008). This first step identifies the origin of the problem, names the limiting assumptions made, and describes the system to be considered, the boundaries, the interfaces and the participants. From this, a problem statement is derived, which describes the initial state, as well as the desired state (Avramenko & Kraslawski, 2008; Baker et al., 2001).

Next, the decision-maker identifies the requirements of the problem. These define which of the possible solutions to the problem are allowed. In mathematical problem solving, these requirements are called constraints. For example, they can be a mandatory minimum or maximum quantity of a resource (Avramenko & Kraslawski, 2008).

The next step is to set the objectives. For this purpose, ‘key figures’ must be selected (e.g. profit, turnover, costs, size). In addition, the decision-maker has to define whether they should be maximised or minimised or whether a certain value should be achieved. In mathematical terms, this is described as the objective function (Avramenko & Kraslawski, 2008; Baker et al., 2001).

After that, the existing alternatives that are available for selection must be identified. These alternatives have to meet the previously defined requirements, if they do not, the subsequent steps will not consider them anymore. Each alternative has to be described clearly (Avramenko & Kraslawski, 2008).

In the next step, the decision-maker determines the criteria to evaluate the achievement of the defined goal. If the previously defined objective is already a clear key figure, then this step is omitted because there is only one descriptive size (e.g. profit). However, if it is a more complex goal that cannot be simply expressed in terms of one key figure, the decision-maker has to define several criteria, which show how well an alternative has met the goal. An example of this would be if the goal is to expand business in a particular country, then possible criteria could be the number of customers in that country, generated sales and completed orders. According to Baker et al. (2001), the criteria should be:

- Able to discriminate among the alternatives;
- Complete, include all goals;
- Operational, meaningful to the understanding of the decision;

- Non-redundant;
- Few in number;

The various alternatives must then be evaluated with regard to all defined criteria. For this purpose, a value must be defined for each alternative, which it receives for a criterion. This requires expertise and information. After all alternatives have been evaluated, the decision-maker creates a ranking list, which lists the possible alternatives according to their value of the objective function (Avramenko & Kraslawski, 2008; Baker et al., 2001).

After the decision-maker has chosen the best possible alternative, the next step is to validate the solution again to ensure that the alternative actually solves the problem. The final solution should fulfil the desired state, meet the defined requirements and achieve the objectives better than the other alternatives. The background of this validation is that the decision-making tool may have been applied incorrectly (Avramenko & Kraslawski, 2008; Baker et al., 2001). Avramenko and Kraslawski (2008) and Baker et al. (2001) describe this as the last step in the decision-making process. However, Guo (2008) adds one more step, the evaluation and monitoring of the solution. This is intended to capture and analyse the impact of the decision taken and, if necessary, provide feedback and an improved basis for the next similar decision (Guo, 2008).

2.2.2 Location in the CE

This research work deals with the decision-making process, which is located at the end of the linear SC. It forms the transition from the linear SC to the reverse processes, which distinguishes the CE from the linear economy. Figure 2-3 illustrates the location of the decision-making process in the CE. As already explained in Chapter 2.1.2, there are two possible ways for an EoU product, if the consumer decides not to use it anymore. One is disposal, which means that it does not return to the SC, which reflects the “make-take-dispose” approach of the linear economy. The second possibility is to take it to a collection point. This is where decision on further steps takes place. The options are also disposal, which the CE tries to avoid, recycling, remanufacturing, repair or reuse.

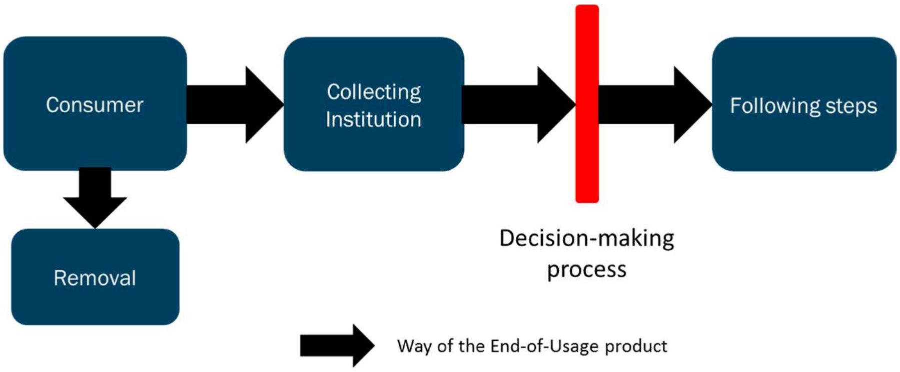


Figure 2-3: Decision-making process in the CE

2.2.3 Decision-making factors in the CE

The decision about the next step of an EoU product is influenced by different factors, the so-called decision factors. A distinction must be made whether this is a strategic programme decision, for example whether a certain product group or brand in general are eligible for CE processes or whether an operational decision is to be made for a specific physical product. Certain information is collected and then a decision is made as to whether it should be reused, repaired, remanufactured or recycled. This research work focuses on the decision on the further steps of an EoU product. Goodall et al. (2014) divide these decision factors into three main groups: social, environmental and economic. Figure 2-4 shows the overall breakdown of these factors.

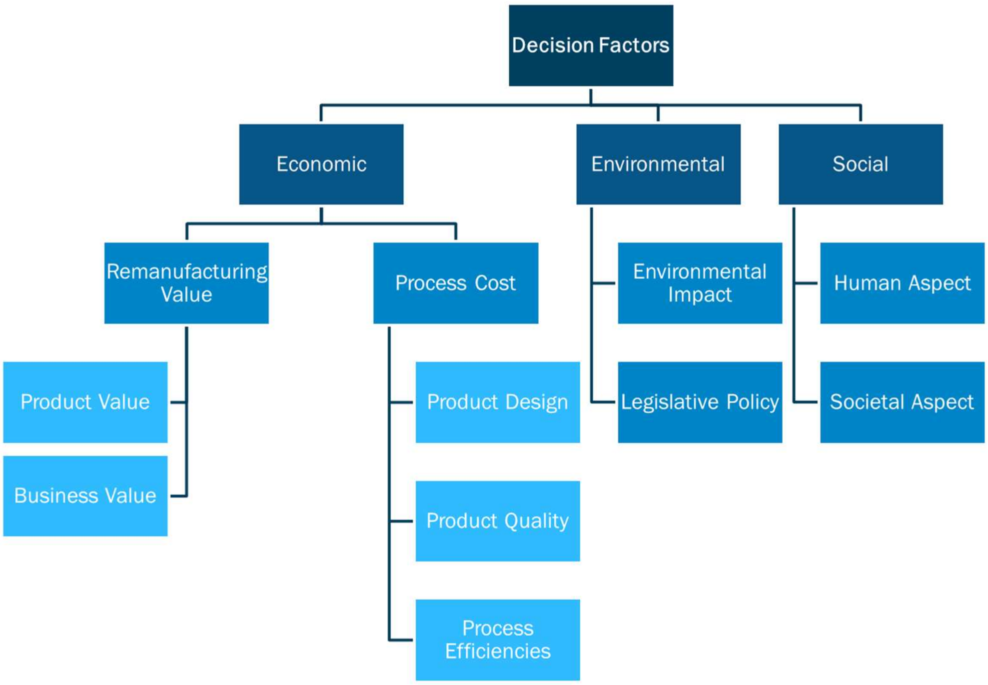


Figure 2-4: Decision factors in the CE. Own illustration based on Dunmade (2004); Goodall et al. (2014); Gurita et al. (2018)

2.2.3.1 Economic factors

The economic factors are further subdivided into remanufacturing value and process costs.

Remanufacturing value: In this case, the remanufacturing value is defined as the possible sales price of the remanufactured product and this determines, multiplied by the sales quantity, the turnover of the product.

Product value: It describes the direct value generated by the individual product. It has a very large impact on the decision of an object. The selling price is strongly determined by the benefit of the product for the customer, which is different for each customer. The benefit depends on the customer's specific requirements and perception, as well as on the product's quality, functionality and available offer (Zeithaml, 1988, p. 14). In addition, the benefit for the customer may increase if the lead time of a reprocessed product or component is shorter than waiting for the completion of a new product (Goodall, 2014, p. 14; Simchi-Levi, Wu, & Shen, 2004, p. 538).

Business value: This value describes the indirect effects on the company's entire business. These are decisive for the strategic orientation of the company. For example, the offer of refurbished products can have a negative impact on the sales of the new products due to the cannibalisation effect (see Table 2-3 challenge 6.). In addition, there may be a loss of reputation if the quality of the reprocessed products is lower than that of new ones and the perceived quality of remanufactured products tends to be less than the quality of a newly manufactured product (Goodall, 2014, p. 10). This is probable when the reprocessing takes place at a third-party remanufacturer (Seitz, 2007). However, it can also have a positive effect as new customer segments can open up (Ellen MacArthur Foundation, 2013, p. 72) or the offer has a positive effect on the product portfolio and supports the sales of other products (Hildebrand, 2005). This value is not decisive for a single decision on a product. It influences programme and strategy decisions.

Process cost: The counterpart of the remanufacturing value is the process cost. This determines how much it costs to bring the product into the desired condition, which depends on which way is chosen by the decision-maker. These costs are based, in addition to the desired quality, on the design of the EoU product, its current quality and the process efficiency.

Product design: The product design influences the process costs, as this affects time, materials and expertise required for reprocessing (Sundin & Bras, 2005). The decisive factor is how easy it is to locate faults and damage, disassemble the product and replace individual components and then reassemble the product and clean it (Lieder, Asif, Rashid, Mihelič, & Kotnik, 2017; Sundin & Bras, 2005). Here a high modularity of the product is of advantage, as work steps are more uniform and spare parts and equipment are more often used. The transportability of the product also plays a role here, as it has to get to a collection institution or remanufacturer and causes additional costs if it is damaged (Parajuly & Wenzel, 2017, p. 279).

Product quality: The product quality describes the current condition of the EoU product. This is used to determine the difference between the target state and the actual state. The larger this difference is, the higher the processing cost (Jun et al., 2007, p. 4581). This is because more steps require more time and more replacement components are used, because less of the EoU product is used. On the other hand, it is more expensive to buy a product core in high quality condition, because it has a higher functionality. In addition to the actual quality, the purchasing price of a core is also determined by the original product selling price and the current demand (Östlin, 2008, p. 98).

Process efficiency: Process efficiency in the CE has an effect on process costs similar to the linear economy processes. The more efficient the processes, the lower the costs. These include low scrap rates, consumption quantities (labour time and materials), short transport routes and automation. In general, avoiding unnecessary or incorrect processes leads to lower processing costs. A big lever in the CE is the pre-evaluation and preselection, since unnecessary transport, purchases and disassembling of EoU products are avoided (Östlin, 2008, p. 45; Tilton, 2013, p. 14).

2.2.3.2 Environmental factors

The second main group of decision factors relates to the environment. The environmental aspect, as shown in the previous sections of this work, plays a very important role in the context of sustainability in the CE. This group includes the impact of the decision on the environment, as well as the legal regulations.

Environmental impact: The business of the manufacturer interacts with their environment and thus has an impact on it. This starts with the fact that resources are required for the production of products that must first be mined, cultivated or partially processed. Certain resources and materials are only available in limited quantities. Therefore, the aim is to use as little of them as possible (European Commission, 2017, p. 4). In addition, some mining or generation methods have a significant negative impact on the environment. Destruction of the environment through rotting or fracking, as well as the release of toxic substances or environmentally harmful emissions, is dangerous (Ellen MacArthur Foundation, 2013). In addition, energy is required for transport between the individual processing stations and for the processing procedures, which consume resources and generate emissions. The reuse of entire products or at least their components can often lead to lower energy consumption than the production of a new product, since certain production steps are omitted. In addition, lower quantities of resources are used (Goodall, 2014, p. 1; Matsumoto & Ijomah, 2013, p. 390). Normally, direct reuse has a lower impact on the environment than repair, remanufacturing and recycling (see Section 2.1.2). However, the effects on the environment that arise during the use phase of the product have not yet been taken into account. For example, the reuse of an old combustion engine may have a more negative impact on the environment than the production of a new combustion engine with a significantly higher energy efficiency, because the new engine uses less energy when it is used. If the same combustion engine is manufactured without technical progress, and thereby retains the same energy efficiency, the reuse is preferable. The energy consumption during the use phase can exceed the required energy for material production and manufacturing (Gutowski, Sahni, Boustani, & Graves, 2011, p. 4541). Since this can also be the case the other way round, that more energy is required during production than during use, deciding which method is preferable must be analysed for the individual product. This is only necessary for products that consume energy, otherwise the reuse will always be better from the environmental point of view.

Legislative policy: The second part of the environmental decision factors is the legislative regulations. These include governmental regulations or agreements by associations. For example, producers can be obliged to take their products back from customers and pay for their disposal or reuse (European Parliament and the Council of the European Union, 2012; Friege, 2012, p. 6). On the other hand, the state can also subsidise forms of reuse, which makes it more attractive for the producers and remanufacturers (Gurita et al., 2018, p. 4). Legislation can also be an indirect obstacle to reuse by limiting attractiveness to such an extent that there is so little or no demand. An example of this are the bans on diesel-powered vehicles in Germany, which make the sale of old diesel powered vehicles very difficult, as they may no longer be driven in some cities (for an example of this in Stuttgart see Stadt Stuttgart (2019)).

2.2.3.3 Social factors

The last main group of decision factors are those related to the social part. Dyllick and Hockerts (2002) differentiate two social aspects of sustainability: the human and the societal aspect.

Human aspect: This aspect primarily concerns the skill, motivation and loyalty of employees (Dyllick & Hockerts, 2002, p. 134). This includes also safety issues for employees, which may be caused by the product or material to be processed, as well as the process itself that has to be carried out. This entails health and ergonomic aspects (Presley, Meade, & Sarkis, 2007, p. 4607). Working conditions and the satisfaction of the employee are in the focus of this aspects.

Societal aspect: This describes the perception of the society concerning the business. On the one hand it can have a negative reputation because repairing and reprocessing used products is not as prestigious as producing the latest high-tech products. On the other hand, a positive image is conveyed due to the increased environmental awareness in society. In addition, the new business segment creates new jobs, although the jobs that are lost in the normal manufacturing segment must not be neglected (Morgan & Mitchell, 2015, p. 20).

2.2.4 Decision stages

Goodall (2014) describes three different stages of decision-making. This starts with the highest level, strategic management, through tactical decision down to the operators assessing individual products. Figure 2-5 shows the hierarchy of decision stages. The structure and principal remains the same in the different stages, but the aims and influencing factors can differ (Goodall, 2014, p. 18).

Figure 2-5: Decision stages. Own illustration based on Goodall, 2014, p. 18

High-level and middle management make the strategic decisions of a company. These decisions have a broad time. With regard to CE, companies decide whether individual CE processes are relevant at all for their business. The strategic alignment of the company is of great importance for this. The decision-maker must analyse and determine the likely impact of the decision on the company's economic viability and image. In addition, they have to consider the required capacities and qualifications. The analyses should also include the external influences due to globalisation, as well as governmental regulations and the demands of society. This is important for an efficient and successful company orientation.

The second level represents the tactical decisions. These tend to be focused towards the medium term. Middle management tries to provide a method for implementing the chosen strategy (Goodall, 2014, p. 21). In this step, the decision-maker defines which products and how many of them should be supplied in the next period to the individual CE processes, as defined beforehand by strategic management (Ferguson et al., 2011). Ferguson et al. (2011) describe this as the 'disposition decision'. This decision is important, because required resources, like certain spare parts are not on sale permanently, but often only at one date for a longer period of time or even for the whole need for a possible returning product in future (Ferguson et al., 2011, p. 774). In addition, the responsible persons must align the capacities, such as workforces and machines, for the planned activities. Since the operational decision-making process about a return product is very cost-intensive, guidelines and specifications are defined at the tactical level (Goodall, 2014, p. 23). According to these tactical decisions, the operational decisions about a concrete returning product are made.

The lowest level of decision-making concerns the operational decisions. This type of decision evaluates individual returning products. This is, besides the time horizon, the main difference to the strategic and tactical decisions, which do not deal with individual products, but with groups and types (Goodall, 2014, p. 24). At the operational level, the decision-maker decides on the basis of previously defined key figures whether a product should be rejected or not. If it is not rejected, a decision is made on the best possible process for returning it to the SC. For this purpose, the current condition of the product must be determined and other factors such as demand and availability of capacities taken into account (Goodall, 2014).

2.3 Machine Learning

This part of the chapter deals with the subject of ML. The idea of ML is introduced and different variants are presented. Finally, the possible areas of application and the state of the art are discussed.

2.3.1 The idea of machine learning

The goal of ML is to teach a machine to perform a certain task independently (Richert & Coelho, 2013, p. 8; Simeone, 2018, p. 2). These tasks are not simple recurring steps that do not require any information, but rather making decisions, creating forecasts or identifying patterns, for which given information needs to be processed (Backhaus, Erichson, Plinke, & Weiber, 2016). For this purpose, the machine analyses a very large amount of data, the so-called ‘training data’. Based on this training data, the machine learns the tasks and can then carry them out independently (Backhaus et al., 2016; Sathya & Abraham, 2013, p. 34). A machine can process much larger amounts of data than a human can. A further advantage is that it can carry out very monotonous tasks reliably and there is no fatigue. In addition, some ML methods are able to analyse unstructured data, such as images, audio files or videos (Manoj krishna, Neelima, Harshali, & Venu Gopala Rao, 2018; Marin, Alféres, Córdova, & González, 2015).

2.3.2 Classification

Classification is a process for predicting qualitative outputs. Qualitative variables are referred to as categorical. This means that the variable can have a limited number of values, in contrast to metric data, which can have infinite different values (James, Witten, Hastie, & Tibshirani, 2017, p. 127). First, this section introduces briefly the methods of decision trees, Naïve Bayes, K-nearest neighbours and support vector machines. They are not discussed in detail, as they do not play a role in the further course of this work. Then logistic regression and ANN are introduced in more detail.

Decision trees: As the name suggests, decision-making is structured here like the shape of a tree. The node at a branch represents the test of an attribute, where it is decided which way to follow. It goes through so many branches until a leaf is reached which represents a certain label (Grossi, Pedreschi, & Turini, 2016, p. 27; Moret, 1982). The determination of the shape of the tree and tests on at the nodes is done based on some heuristic measures (Grossi et al., 2016, p. 27; Quinlan, 1986).

Bayesian Approaches: These are based on Byes theorem and are robust to noisy data and are able to isolate irrelevant attributes (Grossi et al., 2016, p. 28). This is the reason why they are often used when the relationship between attributes and label cannot be determined. Popular approaches of Bayesian classification are naïve Bayes and Bayesian networks. Naïve Bayes classifier assigns an object to the label to which it most probably belongs. Therefore, it assumes that the attributes are conditionally independent. A downside is that continuous attributes must be prepared to compute (Grossi et al., 2016, p. 28).

K-nearest neighbours: This classifier determines the label based on selected neighbours of the object under consideration. The k defines the number of neighbours used. To determine the neighbours, a measure must be selected to determine the proximity of every object (Grossi et al., 2016, p. 29). The method belongs to the lazy learners. This means that no concrete model is trained. Each object under review is compared to every training data set. This shifts the computing effort from the training phase to the application phase, which makes the difference to the eager learners (Grossi et al., 2016, p. 27).

Support vector machines: This classifier separates the set of training objects by means of hyperplanes (Evgeniou & Pontil, 2001). Therefore, the object-free area around this boundary should be as large as possible in order to separate the objects in the best possible way. In order to process data which cannot be separated by linear hyperplanes, this data is transformed into a space in which linear boundaries can do this (Grossi et al., 2016, pp. 29–30).

2.3.3 Logistic regression

Logistic regression forms a development from the linear regression. This section deals with multiple logistic regression, which describes a logistic regression with more than one independent variable (James et al., 2017, p. 135). The logistic regression solves the problem of linear regression, which is that it has an unlimited range of output. Logistic regression provides only values between 0 and 1, which makes it possible to use it for classification, since it can produce binary responses. For this, it calculates a probability that the dependent variable is true. If the probability is bigger than the threshold, the dependent variable is true, otherwise false. In logistic regression, the independent variables are multiplied by regression coefficients to calculate the probability of the dependent variable. This is analogous to the linear regressions, as shown in equation (2.1) (James et al., 2017, pp. 130–132). The regression coefficients (β_n) describe the influence on the dependent variable. However, the value of the regression coefficients cannot be used alone as a measure, since it also depends on the values of the independent variables (James et al., 2017, pp. 130–132). There are variables which can have only the values 0 or 1, as well as variables which can have very large values.

$$p(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n \tag{2.1}$$

The formula for the linear regression is inserted into the logistic function, as shown in formula (2.2) (James et al., 2017, p. 132), so that the output range is between 0 and 1. In order to be able to process independent nominal variables, dummy variables must be created. Each value of the independent nominal variable is treated as an independent variable. This means that a separate regression coefficient is determined for each value of the independent nominal variable (James et al., 2017, p. 85).

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n}} \tag{2.2}$$

At the beginning, the regression coefficients are unknown. They are determined by analysing the available training data. For this, the logistic regression uses the method of maximum likelihood. The intuition behind this is that the coefficients are determined in such a way that the predicted probabilities for each dataset is as close as possible to the right output. The coefficients should maximize the likelihood function shown in equation (2.3) (James et al., 2017, pp. 133–134).

$$l(\beta_0, \dots, \beta_n) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} \left(1 - p(x_{i'})\right) \tag{2.3}$$

The product of all calculated probabilities for which the correct output is 1 should be as close as possible to 1. The product of the calculated probabilities for which the correct output is 0 should be as close as possible to 0, so that the maximum of this function shows the best estimated coefficients (James et al., 2017, pp. 133–134).

The developed model can then be used for predicting the output of a dataset. For this purpose the determined coefficients, as well as the given values of the independent variables, are inserted into the logistic equation (2.2). The calculated probability is compared with the defined threshold in order to determine the output.

2.3.4 Artificial neural network

The theory of ANNs has been developed early. However, due to a lack of computing power on the hardware side, it was not possible to exploit the potential for a long time (Simeone, 2018, p. 1). This section elaborates the necessary theory to apply ANNs.

2.3.4.1 Biological background

The artificial neural networks reproduce the function of biological neurons in the human brain. There, the neurons are the most important components of the nervous system. There are about 100 billion neurons in a human brain, which simultaneously process information (Kruse et al., 2015, p. 8). The cell core of a neuron, which is called the 'soma', has dendrites as input connections and 'axons' as output connections. A neuron transmits its information about its own axon to the dendrite of the downstream neuron (Backhaus, Erichson, & Weiber, 2015, p. 296). Figure 2-6 illustrates the structure and the components of a biological nerve cell.

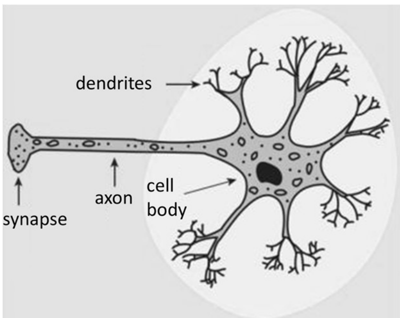


Figure 2-6: Components of a biological nerve cell (Backhaus et al., 2015, p. 297; Kandel et al., 2013)

The communication between neurons takes place by the axon releasing neurotransmitters, which affect the membrane of the receiving dendrite and changes its polarization (Kruse et al., 2015, p. 9). This happens by changes in concentration of negative ions between the inside of the cell membrane and its outside. There are synapses which reduce the potential difference and some which increase it (Kruse et al., 2015, p. 9). Since neurons receive signals from many upstream neurons, the electric potentials accumulate on the cell body (Kruse et al., 2015, p. 10). If a cell nucleus reaches a certain threshold value due to the incoming signals, the neuron generates a short electrical impulse, which is transmitted to the next neuron. By adapting the connections between the nerve cells, learning processes take place in the human brain (Backhaus et al., 2015, p. 296).

2.3.4.2 Different types of artificial neural networks

Artificial neural networks differ in their learning process and their application tasks. The following three learning processes exist (Haykin, 2009, pp. 34–37):

1. *Supervised learning:* It describes learning with a teacher. Here the teacher knows the correct answer, which represents the optimal solution. The training data is therefore labelled. The artificial neural network calculates a solution for an input. An error value is determined on the basis of the teacher's correct solution. It represents the difference between the calculated and the right answer. An error function is set up for which the minimum is searched. For this, the ANN changes settings by using the gradient method so that the value of the error decreases. This describes a movement on the surface of the error function in the direction of a valley or local minimum. These steps are repeated in many iterations until the ANN can satisfactorily determine the solution.

2. *Reinforcement learning:* This describes learning without a teacher. The ANN only gets feedback on the result in the form of a target size. For a classification task, this is the information as to whether the assignment was right or wrong, but not which solution would have been right. Thus, it is not possible to calculate how the network has to be adapted so that it finds the correct answer more probable. However, through many iterations a pattern can be found in random improvements or correct solutions. This has the consequence that in case of reinforcement learning, the network can be trained to be better in carrying out the task than a teacher would be. In comparison to supervised learning, where the network can at most become as good as the teacher.

3. *Unsupervised learning:* This also describes learning without a teacher. However, there is no feedback for the calculated solutions here. Again, unlabelled datasets are used. The ANN should recognise certain patterns in the input data and then create groups or

categories based on these. This is called clustering. The result is that common features in the data are uncovered.

2.3.4.3 Structure and function of an artificial neural network

This section deals primarily with the structure and function of a multi-layer perceptron applying the backpropagation algorithm, as this is the most significant supervised feedforward ANN in practice.

Biochemical processes perform the information processing in biological neural networks. In ANNS, information processing is performed by suitable mathematical computational operations (Backhaus et al., 2015, p. 298). An ANN also consists of several artificial neurons, which are usually arranged in layers. The first layer is the input layer. It is the first interface to the user and receives external information. The neurons of the input layer process the signals and transmit the refracted information to the next layer, the so-called hidden layer. Several hidden layers can exist in one ANN (Backhaus et al., 2015, p. 298). The last hidden layer forwards its information to the output layer. This is the second interface to the user and provides the processed information. Figure 2-7 illustrates the structure of a feedforward ANN. The arrows represent the information, which is transmitted from the input layer trough the hidden layers to the output layer.

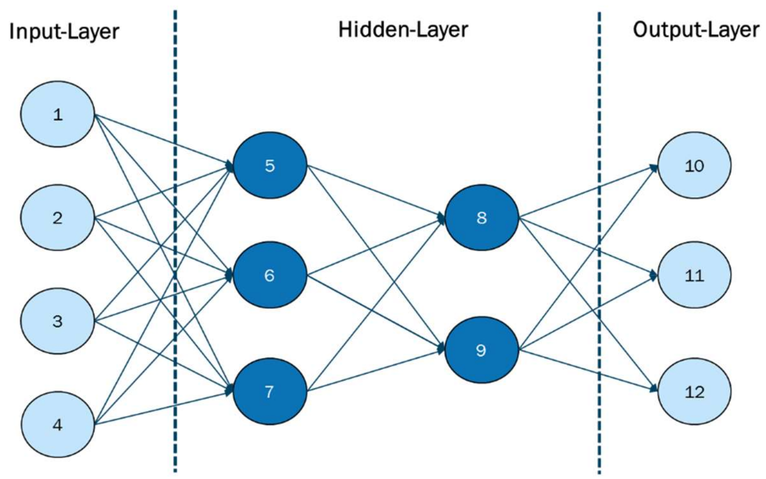


Figure 2-7: Basic structure of an artificial neural network. Own illustration based on Backhaus et al., 2015, p. 299

Feedforward networks cannot use an already calculated result in a further iteration. Feedback networks, on the other hand, can also incorporate old states of the output layer into the calculation by allowing feedback (Backhaus et al., 2015, p. 299).

The information processing in an artificial neuron consists of several steps. Figure 2-8 is an example of information processing for a single artificial neuron.

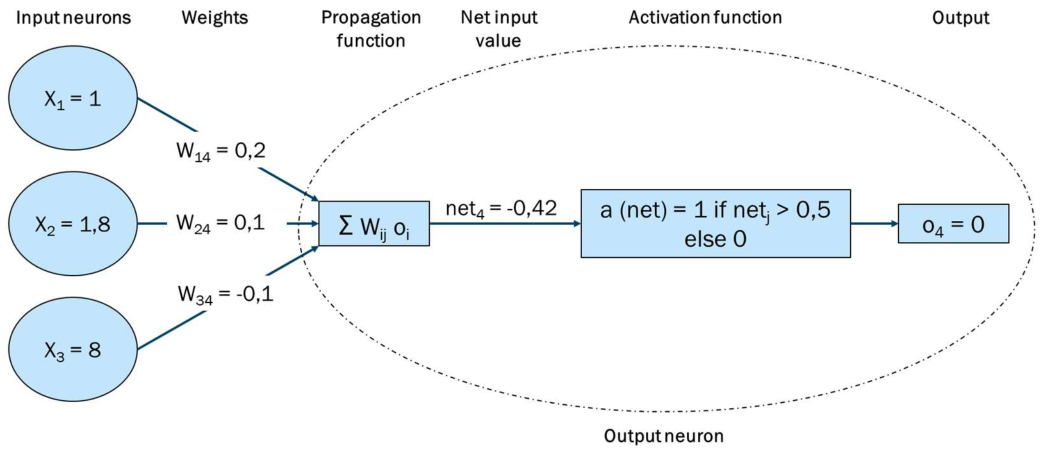


Figure 2-8: Information processing in an artificial neural network. Own illustration based on Backhaus et al., 2015, p. 304

First, all input values that are fed to the neuron are multiplied by an associated-weighting factor, which can be different for each neuron-neuron connection. This can be either external values of input variables, if the neuron is in the input layer; or the output of preceding neurons, if it is not in the input layer (Backhaus et al., 2015, p. 302).

Subsequently, the propagation function processes the weighted input values into a net input value. The propagation function can be for example a sum or a multiplication function. The neuron of Figure 2-8 uses a multiplication function. An advantage of the sum function over the multiplication function is that it can also process an input value of zero. With the multiplication function, a zero already causes the result to become zero, no matter what other input values occur from other upstream neurons. The calculated net input value is processed in the activation function and thus determines the output value of the neuron. There are different activation functions, which are suitable for different tasks (Backhaus et al., 2015, p. 314). Figure 2-9 shows four frequently used activation functions.

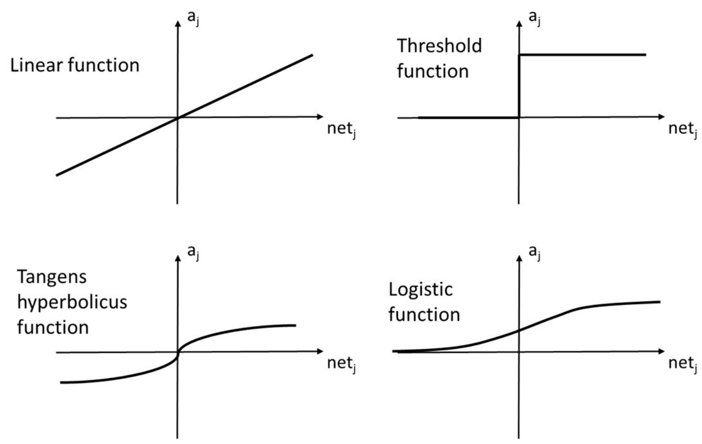


Figure 2-9: Different activation functions. Own illustration based on Backhaus et al., 2015, p. 313; Kruse et al., 2015, p. 45

The linear function has no threshold and there is an existing gradient everywhere, which is important to optimise the values on the basis of the derivation. The threshold function approaches the biological model the closest, since it only distinguishes between activated and not activated. However, it is not constant and therefore cannot be derived. Derivation is a prerequisite for some learning algorithms. Therefore, continuous functions such as sigmoid functions are often used. The advantage of the logistic function over the tangens-hyperbolicus function is that it provides only positive values (Backhaus et al., 2015, p. 314). In addition, it comes very close to the results of the threshold function.

The ANN learns the ability to carry out a task independently through analysing training sets. For this purpose, the neural network receives the input data of a training dataset, as well as the correct output. The neural network independently calculates an output value. The error function calculates the error between the correct output and the output determined by the ANN (Backhaus et al., 2015, p. 317). For the weights in the neural network, a propagation of the error from the output layer to the input layer is assumed. Accordingly, the change of the weights is done backwards. The derivation of the error function is determined in order to train the ANN. It then uses this derived error function to adjust the weights in the network so that the error value is reduced. The gradient method is used to determine the direction and the amount of the weighting change (Backhaus et al., 2015, p. 317), which this work will not discuss further as this exceeds the scope of this work. This is conducted in many iterations until a satisfactory result is achieved.

2.3.4.4 Procedure for implementing an artificial neural network

The following section describes an application-oriented approach for implementing an ANN. Figure 2-10 shows the different steps. The work focuses again on feedforward ANNs. The first step defines the problem the ANN should solve. A network type is selected on the basis of the problem, which usually also defines the used learning algorithm. The available data and their structure and type are also relevant for the network type selection.

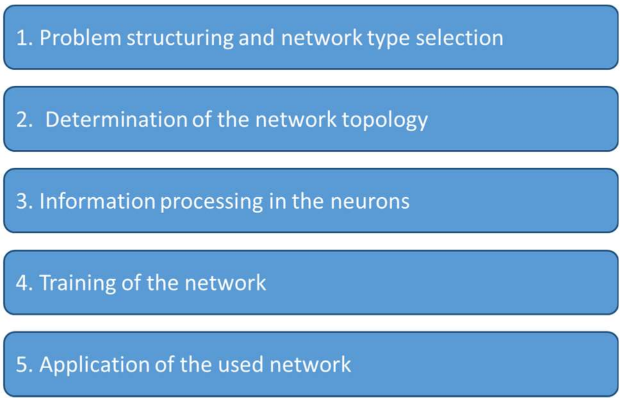


Figure 2-10: Procedure for implementing an artificial neural network. Own illustration based on Backhaus et al., 2015, p. 308

Multilayer perceptrons and radial basic function networks are able to create prognoses and analyse cause-effect relationships. Kohonen maps and hopfields, on the other hand, can determine classes for unlabelled data, which is also called clustering (Backhaus et al., 2015, pp. 307–309). The bigger the artificial neural network is, the more computing power, training data and therefore training time is required (Alsmadi, Bin Omar, & Noah, 2009, p. 378; Backhaus et al., 2015, p. 309). The variance of a variable describes its influence on the output value. If the variance of an independent variable is very small, it can be assumed that it cannot contribute to the parameter estimation and can therefore be neglected (Backhaus et al., 2015, p. 309). The second step determines the network topology. This includes the number of hidden layers, the number of neurons in each hidden layer and the connection structure between the neurons. No firm statement can be made in advance, about which the perfect topology is for the problem. A trial-and-error process can work out a satisfactory solution (Backhaus et al., 2015, p. 310; Sathya & Abraham, 2013, p. 37). However, it should be noted that more neurons and connections increase the complexity. As described before, this leads to an increase in computing power and training duration, as well as the need for more training data. In addition, if the complexity is too high, the risk of overfitting occurs. This happens when the training data is analysed in such a complex way and the network is thus trained very well on the training data, but also learns special characteristics of the training data, which do not occur in testing data. This leads to a worse result on testing and application data (Backhaus et al., 2015, p. 310).

The next step defines the information processing within the single neurons. This includes the propagation and activation function described in Section 2.3.4.3. The selection is based on the given input data and the desired results. In step four, the ANN learns the relations which exist in between input and output values. Through the learning process, the parameters of the neural network are modified in such a way that in each step they improve the ability to reproduce the solution of a problem (Backhaus et al., 2015, p. 314). Theoretically, the user has the following possibilities to influence the learning process (Backhaus et al., 2015; Zell, 2003).

- Development of new connections and deletion of existing connections
- Adjustment of the threshold value of neurons
- Modification of propagation or activation function
- Creation of new neurons and deletion of existing neurons
- Adjustment of weights of connections

In a network in which all neurons are interconnected, the first possibilities are to be seen as adjusting the weightings of the connections. Deleting a connection means that the weight gets the value zero and developing a connection means that the weight is no longer zero. There are currently no algorithms for the modification of propagation and activation functions by which this step can be performed (Backhaus et al., 2015, p. 316). The development and deletion of new neurons usually takes place after the weights in the network have been optimised and the result is still not satisfactory. The modification of the threshold values can be recorded as modifying the weights. The focus of the learning process is therefore on adjusting the weights of the connections between the neurons (Backhaus et al., 2015, p. 317; Sathya & Abraham, 2013, p. 34). After the ANN has adjusted its weights based on the training data, the performance of the trained network is tested. The active learning process is terminated and the network generates the output for input data, for which the output is already labelled. Afterwards, how reliably the trained ANN can calculate the output values in the midfield state is analysed. If the results are not satisfactory, it will be trained again with new starting parameters, otherwise the learning phase will be completed (Backhaus et al., 2015, pp. 321–323). The last step is the application of the trained ANN.

2.4 State of the art

This section serves to examine the state of the art of decision-making regarding the different recovery options. The first step is to identify the stage of the decision-making process. The process considered in this work is at the operational stage, as a decision is made regarding individual laptops (see Section 2.2.4).

Goodall et al. show in their study from 2014 the current literature regarding remanufacturing evaluation decision tools. There is a clear difference between the individual decision stages in terms of the number of literature. Table 2-4 shows the quantity of literature for every decision stage.

Table 2-4: Literature in the field of remanufacturing evaluation decision tools

Stage	**Number of literature**	**Use of economic decision factors**	**Use of ecological decision factors**
Operational	4	4	0
Tactical	22	18	15
Strategical	15	11	10

This suggests that the decision on recovery options is made for entire product groups or products, but not individually for specific products that could be recovered. This is also reflected in the fact that buyers of used products often only buy functional products. As soon as the device no longer works, it is not purchased anymore. It does not matter how many components are defective. Only the condition of the battery is relevant, because it is the best known wearing part and the replacement is done with low risk.

A further difference can be seen when looking at the decision factors used in the remanufacturing stages. The focus of the operative evaluation is on the economic factors. The ecological factors are only taken into account at the higher levels.

Table 2-5 lists four research projects in the field of remanufacturing evaluation decision tools on the operative stage. Two of them are included in the literature provided by Goodall et al. (2014).

Table 2-5: Literature with similar focus to this work

Reference	Title	Product physically present
Goodall et al. (2015)	Cost estimation for remanufacture with limited and uncertain information using case based reasoning	Not defined
Jun et al. (2012)	Heuristic algorithms for minimising total recovery cost of end-of-life products under quality constraints	Yes
Xu et al. (2014)	Cost modelling to support optimised selection of End-of-Life options for automotive components	Yes
Zhou et al. (2012)	A quality evaluation model of reuse parts and its management system development for end-of-life wheel loaders	Yes

Closest to this work is the cost estimation of Goodall et al. (2015). They have developed a method that uses the manufacturer, model and quality to determine the remanufacturing costs. Therefore, a combination of case-based reasoning and probability theory is used. Case-based reasoning is an artificial intelligence tool. For a new individual object, historical data is searched for similar input features regarding manufacturer, model and condition. The corresponding costs are the output. The condition of the product is recorded in five different stages. This does not describe how the condition of the object under review is determined.

Jun et al. (2012) assume that the condition of the product under review is known. Based on this information, they determine the best recovery option for the components. Heuristic algorithms are used to minimise the recovery costs. They refer to the need to predict the product quality in order to use it in their tool.

The cost model developed by Xu et al. (2014) follows the activities-based costing approach. The condition of the product is determined by means of an inspection. Therefore, the product has to be physically present. The activities required to bring the product into the desired condition are derived from the determined condition. These are used to determine recovery costs of the EoL product.

Zhou et al. (2012) developed a model to determine a reusability score for the components. Therefore, the product quality is evaluated through several inspections, which implies that the product is physically present. A fuzzy analytical hierarchical process is then used to calculate the score. The score shows whether it makes sense to reuse the part or not.

Another research work in this field is the one of Ghazalli and Murata (2011), which is located at the strategic level, as it comprises a one-time decision per product type. They developed an analytical hierarchical process using case based reasoning and the nearest neighbourhood algorithm to evaluate different EoL strategies. The process is split in two and determines the EoL option first at product level and then at component level. The components are not evaluated until the nearest neighbourhood algorithm has determined remanufacturing as EoL option for the product. Subsequently, remanufacturing values and recycling values are calculated to determine the EoL strategy for the individual components. Ecological factors are also taken into account.

The paper *Artificial Intelligence and the Circular Economy* by the Ellen MacArthur Foundation (2019) provides an exploration of the intersection of AI and CE. The necessity for an AI-based analytical model is identified in order to enable a decision-making process regarding the recovery, based on condition information of the product under review (Ellen MacArthur Foundation, 2019, p. 14). Furthermore, enabling pricing of used products is highlighted as this is a boosting factor, as it drives people to sell their EoU product and shifts their preference to buying items that hold their value longer. The use of AI is intended to take market conditions and product-specific features into account (Ellen MacArthur Foundation, 2019, p. 28).

AI and ANNs are already used for prediction tasks in the context of diagnosis. For example, they are used in the health sector. Symptoms, information about the patient, laboratory values, pictures and other information are used to make a diagnosis or recommended treatment.

ANNs are used by Marin et al. (2015) in order to detect melanoma through image recognition. The ANN is trained by means of many historical pictures, which are known to either be melanomas or not. The trained ANN can then reliably predict from a picture whether it is a melanoma. The patient does not have to be present.

El Jerjawi and Abu-Naser (2018) use an ANN to determine whether a person has diabetes or not based on input such as age, blood pressure, body mass index and more. This a feedforward binary class ANN, with a backward propagation learning algorithm, where only one label can be accepted at a time.

2.5 Chapter summary

This chapter serves the development of a basic understanding of the important fields, which is necessary for this research work. To this end, the CE, its structure and mode of operations, as well as the possibilities and challenges, are discussed first. The subsequent focus is in the decision-making process. For this purpose, the general decision-making process is examined and the decision factors in the CE in particular. In the next step, the field of ML is examined. The method of classification and thereby the logistic regression and ANNs are discussed. Finally, the state of the art regarding decision-making in the CE is discussed, as well as the use of ANNs for predictions and diagnosis.

Chapter 3
Development of the decision-making process

This chapter serves to develop an improved decision-making process. The findings of the previously conducted literature review are used for this purpose. First, the problems identified in Section 2.1.5 are checked for similar factors. The results of the problem analysis are then used to derive requirements for an improved decision-making process, which is then defined. Afterwards, stress factors, which influence the condition of laptops, and ways how to capture them will be worked out. The ML method and a ML software are selected to perform the first part of the decision-making process, the determination of the condition of the components. Finally, the economic and ecological evaluation of the EoU laptops are developed.

3.1 Problem analysis

Section 2.1.5 identified the current problems of CE. A closer look at the problems reveals that some of them have a common denominator or can be improved by common measures. This is the decision-making process that determines which path an EoU product should take, if a customer decides to return it to the SC. As problem no. 1 shows, the labour costs of recovery options are high as the degree of automation is very low. This is also the case for the decision-making process and therefore labour costs could be reduced by using machines. Another opportunity to reduce labour costs is to avoid unnecessary manual quality checks of returning laptops. A decentralised pre-evaluation, without the laptop already being physically on-site could determine the quality before the customer sends the laptop to a collection point. In addition, the following path of the EoU laptop could be determined in advance and the laptop could thus be sent directly to the appropriate process. This could reduce the number of collection points. That also solves parts of problem no. 5, the missing infrastructure. The short lifetime cycles and product diversity mentioned in problem no. 2 would also benefit from a decentralised decision-making solution with central know-how management. A centrally controlled software is more flexible and can be adapted or extended more quickly. The further training of personnel who carry out the diagnosis at collection points requires significantly longer lead times and is more cost-intensive. A software application can cover many more products. Problem no. 3 describes the uncertainties regarding the return of EoU products. The economic aspects are of particular importance here, as they are a driving force to investment. Ecological factors should also be taken into account. An upstream economic evaluation would reduce the risk and economic uncertainty of acquiring an EoU product that would lead to a loss of money. Many of the products currently in use have not been developed for the CE, as problem no. 7 shows. Therefore, condition monitoring is usually not possible due to a lack of sensors. The available information must be used to determine the condition. No. 8, the last problem, concerns the lack of incentive for customers to return their EoU products to the SC. In most cases, only products that can be resold directly are purchased from companies after a diagnosis of the condition with the physically present laptop. Based on the condition, the owner of the EoU laptop receives a price suggestion. It would be better if the customer were to receive a non-binding offer before sending in the laptop. This decentralised solution would probably reduce the hurdle of returning a product and the customer would not choose the easier way to dispose of the laptop.

3.2 Structure of the new decision-making process

This section develops a structure of the decision-making process for EoU laptops, which is based on the problems analysed in the previous section. This decides which alternative of reuse, repair, remanufacturing or recycling is the best option, with regard to economic and ecological factors.

In order to carry out the decision-making process decentrally, the decision should be made when the laptop is still with its owner. The know-how in the software is centrally located at the software provider. This is comparable to a cloud application. The data in the cloud is accessible decentrally, but the cloud provider stores the data centrally. For this purpose, on the one hand general information about the laptop model will be used, but information about the individual laptop is necessary as well. These data should be combined to predict the condition of defined components by using ML. Ten components were selected for this purpose. These are either typical wear components, sensitive to external influences or have a high value. Therefore, they are the focus of the economic evaluation. The condition of the following components is to be determined: Battery, CPU, GPU, display, keyboard, touchpad, motherboard, housing, hard drive and the cooling system.

This process step describes a mixture of diagnosis and prognosis. Diagnosis aims to determine the root cause of a fault event that has already occurred (Uckun, Goebel, & Lucas, 2008, p. 2).

Prognostics describes the process of predicting a future state or remaining useful lifetime of an object based on its current and historic conditions (Bektas, Jones, Sankararaman, Roychoudhury, & Goebel, 2018, p. 87; Kumar, Sotiris, & Pecht, 2008, p. 3674; Uckun et al., 2008, p. 2). Aspects from a diagnosis are applied because obvious symptoms of the laptop under consideration are used to infer the condition of the laptop. In addition, knowledge from historical data is used to predict the condition of components, as in prognosis. The required information should be easy for the laptop's owner to capture, so that he can enter it independently in a provided application. The determination of the condition of the defined components forms the first part of the decision-making process. The determined conditions of the components are then used in a subsequent evaluation. The condition of the components indicates whether this component must be replaced before it is returned to the SC. It should be noted that there are two types of defects: The patent defects and the latent defects (Kuo, Chien, & Kim, 1998, p. 103). Patent defects are easy to recognise because they appear externally or restrict the functions of the object. Therefore, they can be detected by functional testing or inspection. On the other hand, latent defects are not obviously recognisable until they transform into a patent defect. Products with latent defects therefore often have a significantly shorter remaining useful lifetime. Therefore, latent defects should also be predicted in order to replace these components as well as the defective components (Kuo et al., 1998, p. 103). This would have a positive effect on the remaining service life and thus on the customer satisfaction. Defined information is stored in the system for the economic and ecological evaluation. This information is used to determine the estimated costs that will be incurred to restore the functionality of the laptop. These costs are offset by the expected selling price. On the basis of the difference, a price can be calculated which can be paid to the owner of the EoU laptop, so that it is still economical. If the costs already exceed the expected selling price, the owner is advised to recycle the laptop in order to extract materials from it. In this model it is not defined whether the individual processes are carried out by the same or by a third party.

Figure 3-1 shows the structure of the developed decision-making process for EoU laptops. The first part is the determination of the condition of the laptop components by using model and individual laptop information in an ML process. The second part forms the economic and ecological evaluation, which shows the best alternative.

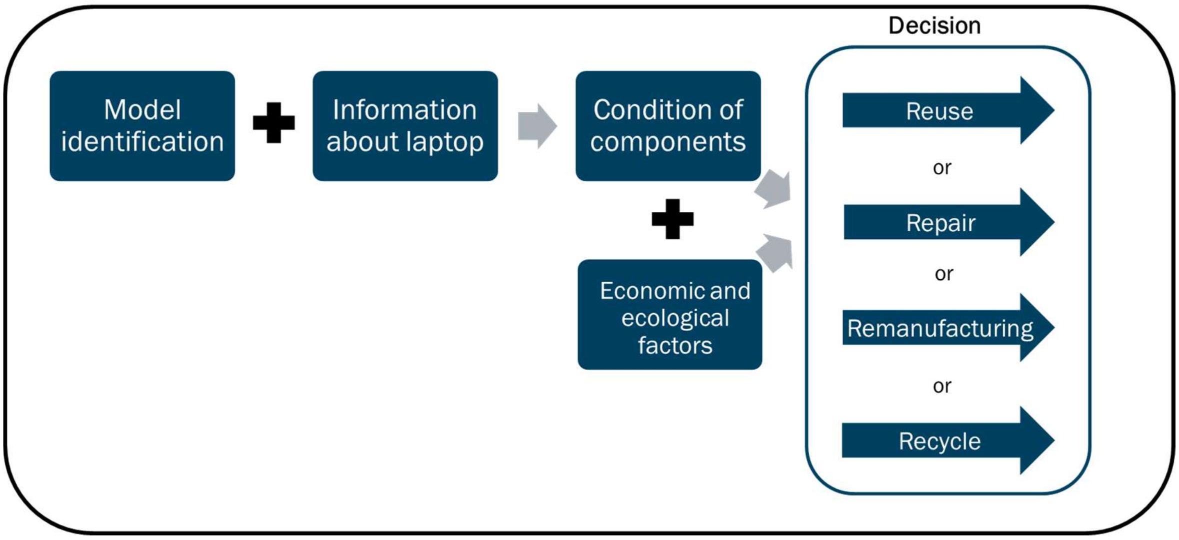


Figure 3-1: Developed decision-making process for laptops

3.3 Stressing factors for laptops

There are different stressing factors to which laptops are exposed during their lifetime. These have an effect on the condition of the laptop and its components. A distinction can be made between environmental and operational stress as Table 3-1 shows (Panasonic, 2012, p. 5). Environmental stress describes influences that the laptop is permanently exposed to, even when not in use. In contrast, operational stress occurs when the laptop is actively used. Since a laptop consists of different components, which are exposed to the influencing factors to different degrees, different influences must be taken into account. Components that are heavily stressed by active use of the laptop include electronical components such as semiconductor elements, input devices such as the keyboard and touchpad, and output devices such as display.

Table 3-1: Stress factors of laptops

Environmental stress	**Operational stress**
Temperature	Temperature rising, local heating
Humidity, moisture	Current
Pressure	Voltage
Dust, dirt	Electric/ magnetic field
Radiation	
Electromagnetic waves	
Shock, fall	
Bending	

The temperature of the laptop components is influenced by two factors. On the one hand, the environmental temperature, which is determined by the location of the laptop. On the other hand, the produced heat of electrical components through usage, which increases the temperature inside the laptop (Panasonic, 2012, p. 5). A higher temperature over a longer period of time leads components and material to age faster (Celaya, Wysocki, Vashchenko, Saha, & Goebel, 2010, p. 2; Squaretrade, 2009, p. 3). In addition, exceeding certain temperatures can cause immediate damage to individual components. The heat problem with laptops is mainly due to the compact design (Haberer, 2009). Furthermore, very low temperatures can have a negative effect on the components due to the formation of condensation. This is particularly the case with a very rapid temperature reduction. Condensation can lead to corrosion of electrical components or to a short-circuit, which destroys the components immediately, analogous to water damage (Patona International, 2019).

Humidity has an influence on electrical components for the same reasons as low temperatures. Humidity leads to moisture, which leads to corrosion. If the humidity is so high that water drops are formed, a short circuit can occur, which destroys electrical components (Kuo et al., 1998, p. 94).

The accumulation of dust and dirt under the keys of the keyboard can lead to a functional restriction of these. More serious, however, is that dust can accumulate in the ventilation part of the cooling system. As a result, the heat energy generated inside the laptop cannot be dissipated in the desired form and the temperature can no longer be controlled sufficiently (Becker, 2015). The excessive temperatures lead to damage of the components, as already explained.

Falling can cause damage to the outer casing of the laptop. This can lead to optical damages as well as functional damage. The display reacts very sensitively to shocks, which often cause it to break. In addition, falls can lead to the breakage of components and connections inside the laptop. The effects of bending are relatively similar to those of falls and shocks (Panasonic, 2012, p. 5).

Environmental pressure, radiation and electromagnetic waves also have an effect on the condition of a laptop (Panasonic, 2012, p. 5). However, these factors are not discussed in detail in this work, as it is assumed that an average laptop is not exposed to large pressure differences, strong radiation or strong magnetic waves.

Besides the increase in temperature due to operation of the components that have already been mentioned, the components also age due to the occurring current and voltage. In addition, high current leads to high temperatures. These aging mechanisms include Bias Temperature Instability, Hot-Carrier Injection, Time-Dependent Dielectric Breakdown and Electromigration (Kraak et al., 2018, p. 1). This work does not discuss these in more detail as they exceed the scope of this work. However, it can be stated that active use of the laptop generates currents and voltage, causing the components to age.

The decisive determinant for these factors is how long they have been present for and how badly they affect the laptop. For the environmental stress factors, age is the defining time value. The influence of operational stress factors on the product results from the usage time and the usage intensity. In addition, the quality of the entire laptop and the components used is also relevant for ageing. High-quality materials and components do not age as quickly as cheap ones and therefore have a longer service life (Squaretrade, 2009, p. 1).

3.4 Ways of recording stress factors

Some of the above-mentioned stress factors cannot be recorded directly or can only recorded with difficulty. Therefore, indirect variables are also used to capture the influence of stress factors, since they can provide information about them.

The quality is reflected in the brand, price and model of the laptop. The model also defines the design of the laptop. This has an effect on the protection and configuration of components as well as on the ability to dissipate heat.

The usage time describes how long the electronic components were exposed to current and voltage, which leads to aging of the components, as explained above. It also provides information on how often mechanical components have been stressed, depending on the type of use. In addition, the recorded time of usage can deviate from the actual time of usage, if external devices were used. These could be an external screen, mouse or keyboard (Becker, 2015).

The battery of a laptop is one of the parts that quickly wears out and reduces load capacity. Besides the age, this is mainly due to the charging cycle (Patona International, 2018). In addition, the number of charging cycles or the total charging time also provide information about how much the laptop was used. This information would then be more relevant than only the age of the laptop.

As described above, temperature is one of the most important factors influencing the condition of a laptop. Normally, laptops have an integrated temperature sensor, which measures the temperature inside the laptop and uses it to regulate the cooling system. The temperature of individual components can be checked manually in the laptop system. The implementation of recording software should make it possible to track the temperature curves. This would provide information on how much and how intensively individual components were used and how long high temperatures stressed the components. Maximum and minimum temperatures would also be interesting. The maximum value would indicate whether a limit value was exceeded above which parts would be directly damaged. The minimum value would show whether water drops could have formed. This would be a consideration related to the past. On the basis of these historical values, the condition of the components would then be estimated. On the other hand, the temperature can be used in monitoring. An increase in temperature, while the needed computing power remains the same, would indicate that the cooling system or one of the components used is defective.

The load of the CPU and the GPU can be recorded in a similar way to the temperature in the laptop system. These values provide information about how much the unit is used. By using suitable software, these load curves could be recorded. High utilisation leads to faster wear, as it describes the intensity. In addition, higher loads normally result in high temperatures, which in turn leads to accelerated aging. The degree of utilization is primarily determined by the tasks to be performed by the laptop. Computationally intensive programs such as calculations or gaming lead to higher utilisation. The type of use of the laptop can thus be defined, if the load of the components cannot be recorded.

For the detection of moisture, as well as acceleration to detect falls, there are normally no sensors in a laptop. Accordingly, this hardware would have to be installed afterwards. The moisture could be recorded permanently to get information about possible corrosion damage. Maximum values could also be recorded. To detect falls, it would be sufficient to store values above a certain threshold, which would indicate a fall.

The extent to which dust and dirt have accumulated in the fan of the laptop cannot be measured easily. However, laptops in mobile use are exposed to more dirt and dust. This information can be easily captured. In addition, a case protects a laptop from dust and dirt. This information can also be captured easily. Mobile use and the use of a case also have an impact on the occurrence of fall damage. Laptops in mobile use tend to have a higher risk of falls. The case, on the other hand, can prevent or reduce damage caused by falls (Becker, 2015). The noise of the fan of the cooling system can also be detected. If these differ from those of a device in mint condition, a large accumulation of dust can be inferred.

General use marks, as well as external damages can be recognised through a visual analysis. Scratches, larger marks and even broken parts can be detected. This visual inspection can be carried out by a human being or by a machine. External damage could indicate damage inside the laptop. In addition, many scratches can indicate poor handling of the laptop or a high degree of usage. This in turn also influences the condition of the internal components.

The data that provides information about the condition of a laptop can be divided into two groups. This is done on the basis of the time at which the information is collected. The first group comprises the information that is gathered at the stage when the product is at the decision-making process. This does not require an upgrade of the laptop. Therefore, this information can be collected for most laptops currently in use without being explicitly designed for the CE. Table 3-2 lists this information, as well as the data type of the data.

Table 3-2: Information at the time of evaluation

Information at the moment of evaluation	Data type
Brand	String
Model	String
Price	Integer
Age	Integer
Type of usage (creating categories)	String
Mobile usage	Boolean
Usage with a case	Boolean
External damages (creating categories)	String
Sounds (creating categories)	String

The second group comprises the data that must be collected and stored during the active usage phase. For the time of usage, utilisation of CPU and GPU, temperatures, usage of external devices and number of charging cycles, it is necessary to install a software application on the laptop to read and save the data of the already existing sensors. For the detection of acceleration and moisture, the corresponding hardware must be installed in addition to the software. Since the collection of this information requires an active retrofitting of the laptop or a direct design of the laptop for the CE and this work concentrates on laptops already in use, these factors are no longer taken into account. However, they should be considered for future developments as they would probably significantly improve the ability to determine the condition of the laptop. Installing one software application to store the information from the already installed sensors would require little effort and will probably lead to significant improvement. On the one hand, the manufacturer of the laptop could do this before sale. On the other hand, a customer who is already aware at the time of purchase that he wants to easily return the laptop to the SC at the end of time of usage, could install the software. Therefore, the software has to be installed immediately after the purchase and before the laptop is used, otherwise falsified data will occur. Table 3-3 lists the information that can be recorded during the usage phase.

Table 3-3: Information during the phase of usage

Information during the phase of usage	**Data type**	**Prerequisites**
Time of usage	Integer	Software
Utilisation of CPU and GPU	String	Software
Temperature	Float	Software
Acceleration	Float	Hardware and software
Moisture	Float	Hardware and software
Usage with external devices	Integer	Software
Charging cycles	Integer	Software

Besides the information that has to be recorded during the usage phase, the sounds of the laptop are not used in the model developed in this work either, because it is difficult to collect and evaluate these data decentrally. Therefore, the following data are collected and processed in the ML process to determine the condition of the laptop:

- Brand
- Model
- Price
- Age
- Type of usage
- Mobile usage
- Use of a case
- External damages

External damages are recorded for six components of the laptop. These are (i) the top casing outside, (ii) display, (iii) bottom casing outside, (iv) bottom casing inside, (v) keyboard and (vi) touchpad. This damage is to be recorded in order to infer internal damages. Since artificial neural networks are able to recognise patterns of which the user is not aware, it is possible that certain damage has a high influence on the state of a certain component inside the laptop. For example, a dent on the base could indicate a defective CPU, as it tends to reflect damage if the laptop crashes to this point. Table 3-4 shows the defined input and output of the ANN.

Table 3-4: Input and output of the artificial neural networks

Input
Brand (String)
Model (String)
Price (Float)
Age (Integer)
Type of usage (Categorical/ String)
Mobile Usage (Boolean)
Use of a case (Boolean)
Condition top casing outside (Categorical/ String)
Condition display (Categorical/ String)
Condition bottom casing outside (Categorical/ String)
Condition bottom casing inside (Categorical/ String)
Condition keyboard (Categorical/ String)
Condition touchpad (Categorical/ String)

Output
Condition battery (Boolean)
Condition CPU (Boolean)
Condition GPU (Boolean)
Condition display (Boolean)
Condition keyboard (Boolean)
Condition touchpad (Boolean)
Condition motherboard (Boolean)
Condition housing (Boolean)
Condition hard drive (Boolean)
Condition cooling system (Boolean)

3.5 Selection of ML method

After defining the information used to determine the conditions of the laptop components, an ML method can be selected. For this purpose the insights elaborated in section 2.3 and 2.4 are used.

The output of the ML method should be able to take two different values. The first class describes a component in good condition, which can remain in the laptop. The second class describes components, which have to be replaced, as they are either defective or have a short remaining service life. The method must also be able to process different data types. There are string, float, integer and boolean variables.

Since ANNs are well suited for classification and uncovering unknown correlations, they are selected (Abiodun et al., 2018). The potentials of ANNs will improve with the progress of digitalisation and technologies (Pinegger, 2018, p. 5). The flow of information processing is feedforward. As learning algorithm, the backward-propagation algorithm is selected. This is the most common variant of an ANN for classification purposes. In order to reduce the complexity of the ANN, a single label ANN is selected. This has only one output neuron, which leads to a lower need of training data due to lower complexity. However, this means that a separate ANN must be trained for each laptop component, which increases the computing effort, but the same data sets can be used. Figure 3-2 shows the structure of the ANN for the battery.

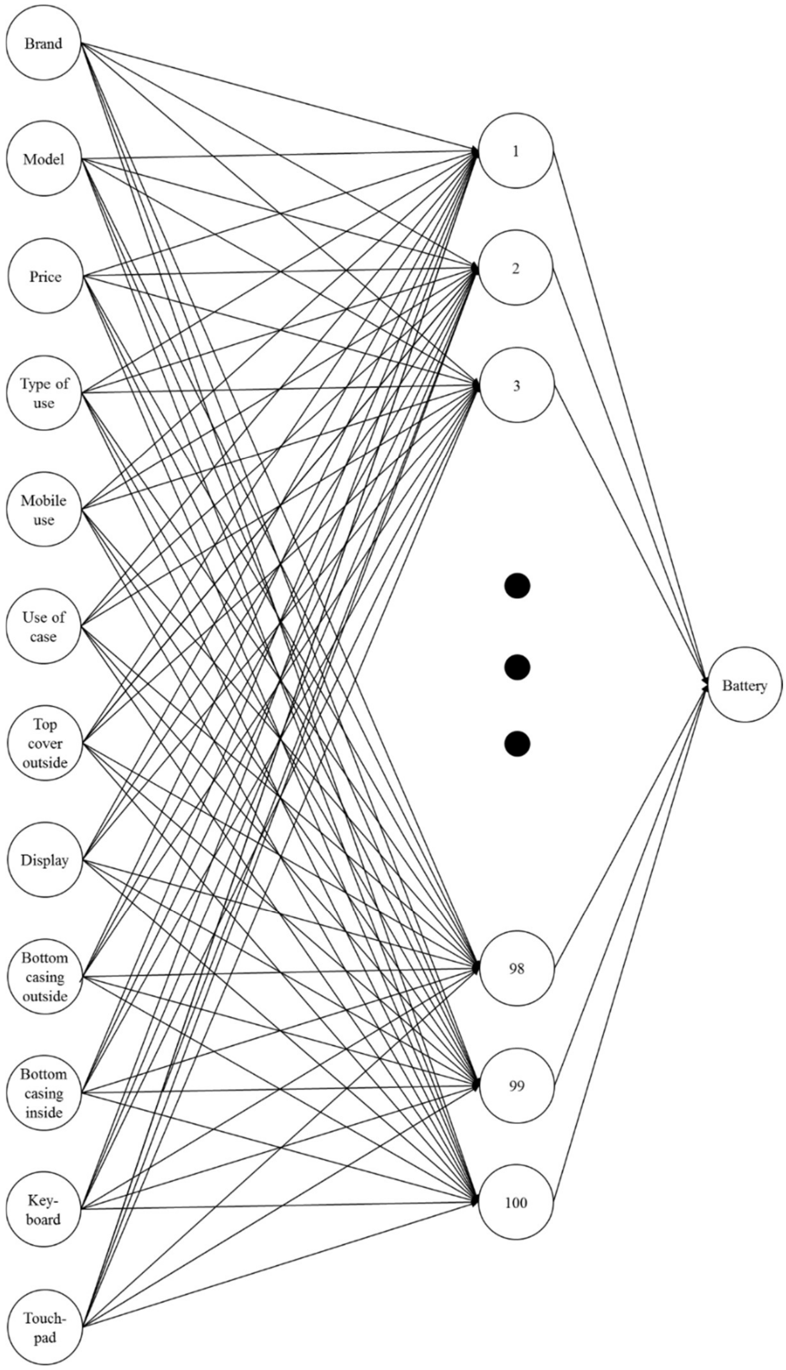


Figure 3-2: Structure of the defined artificial neural network for the battery

In the hidden-layer only six of the 100 neurons are shown. But it is nevertheless already recognizable how many neuron-neuron connections must be trained on the basis of the training data.

3.6 Selection of ML software

This section selects a suitable ML software for the first part of the developed decision-making process, the determination of the condition of the selected laptop components. The software should be able to reproduce the previously defined ANN in the best possible way and thereby fulfil further criteria, which will be developed in the following section. Once the criteria have been determined, weighting factors for the individual criteria are calculated by means of a pairwise comparison. These are then used to evaluate selected software and identify the most suitable one.

The first evaluation criterion is the complexity of the software. This describes how easy it is to work with the software. Influencing factors for this are the user-friendliness, intuitive handling and support information for certain tasks and features (Bhargava, Aziz, & Arya, 2013, p. 309; Jadhav & Sonar, 2011, p. 1397). It is mainly determined by the effort required to implement the previously defined ANN.

Next, the adaptability and expandability will be assessed. This describes whether it is possible to adapt the software to your individual requirements. In addition, it is crucial as to whether the developed application can be extended retrospectively (Bhargava et al., 2013, p. 309; Jadhav & Sonar, 2011, p. 1397). This includes, for example, changes to the network structure or the data to be processed.

Another important aspect is data import and export. Since data is processed in the software but is not generated in the software, it is important that the import is easy and fast. This also includes the data types in which can be imported. Since the knowledge gained by the ANN is further used in a downstream process, it must be possible to export it. In addition, an interface should be established so that external software (in this case Python) can initialise the data import and export. This is also described as interoperability (Bhargava et al., 2013, p. 309).

Support forms the interface to the software provider, who will be contacted in case of technical problems or inquiries. The quality is characterised by the contact channels, response time and the quality of provided solutions (Bhargava et al., 2013, p. 311; Jadhav & Sonar, 2011, p. 1398). This support is especially important in the start-up phase of an application, because this is when the most problems occur, but also during normal operation, because downtimes have a massive impact on the business.

The performance of software comprises several aspects. One of these is efficiency. This describes how well and how quickly the software produces results and how much data it needs to do so. The quality of the result is therefore the most important here. Furthermore, the user interface, handling and the opportunity for visualisation and standard reports also count. In addition, the reliability of the software is a factor to be considered. This includes the robustness, the handling of faulty data and the backup or recovery function in case of a crash. The software should run consistently without crashing (Bhargava et al., 2013, p. 309; Jadhav & Sonar, 2011, p. 1397).

The competence of the software vendor has a major influence on the success of the implementation of the application. This can be determined by the reputation of the vendor, which is made up of popularity, product history and experience (Jadhav & Sonar, 2011, p. 1398). In addition, if you want to apply the software globally, you will benefit by using a global vendor as they have market-specific experience. A financially strong vendor also has the advantage of not being excluded so easily in the event of economic regression. That would be bad, because then the support and updates would no longer be continued and the software could not be developed further. Moreover, a well-known brand could have a positive effect on the marketing of the whole application, as potential customers associate the brand with know-how and success.

Accessibility is closely related to data import and export. This describes how well the software can be accessed and where the data are stored. Because the software is to be used decentrally, it should run in a cloud. Since this research involves a very comprehensive and fast-growing amount of data that are generated decentrally, the data should also be stored in a cloud.

Costs are used as an economic factor. The total costs consist of the licence cost, hardware and software cost, installation and implementation cost, maintenance, training and upgrading costs (Jadhav & Sonar, 2011, p. 1399). The total of the cost blocks and the calculation methods may vary between different providers. For example, there are providers who charge money for the use of their software. Other vendors run the software only on their own hardware and the customer must pay for the hardware used.

In order to determine how much the individual criteria influence the choice of software, they were compared to each other in a pairwise comparison. There are three possible results for each comparison: (i) the criterion is more important than the other one, then it gets three points; (ii) the two criteria are equally important, then it gets two points; (iii) the criterion is less important than the other one, then it gets one point. Figure 3-3 shows the pairwise comparison.

	Criterion	1	2	3	4	5	6	7	8	∑	Weighting	Rank
1	Complexity		3	2	3	3	1	3	3	18	0,16	3
2	Adaptability/ Expandability	1		1	3	1	1	3	2	12	0,11	6
3	Dataimport/ -export	2	3		3	3	3	3	3	20	0,18	1
4	Support	1	1	1		1	1	1	1	7	0,06	8
5	Performance	1	3	1	3		1	3	2	14	0,13	4
6	Competence	3	3	1	3	3		3	3	19	0,17	2
7	Accessibility/ Cloud	1	1	1	3	1	1		1	9	0,08	7
8	Costs	1	2	1	3	2	1	3		13	0,12	5
	3 = more important as					Controlling sum		112		112		
	2 = as important as											
	1 = not as important as											

Figure 3-3: Pairwise comparison of software selection criteria

The individual comparisons are not discussed in detail in here. To determine the weighting factors, all the points one criterion received are summed up and divided by the sum of all points. This value forms the size of the part of the selection. Figure 3-3 shows that the data import and export is the most important criterion. This is due to the fact that this is the basic prerequisite for everything else. In addition, this often plays a role due to permanent data transfer. Second place is taken by the provider's competence, as this combines many partial criteria. This ensures that the vendor has experience in this field and that the implementation of the application is less risky. Next comes complexity. This should not be too high, so that it is possible to quickly familiarise oneself with the application and use it. The ANN should not be programmed from its very beginning, but only specifically adapted. The fourth most important criterion is the performance of the software. It should be well structured and efficient. The learning algorithm should train the ANN with as little data as possible. In addition, the processing time of the training datasets and the calculation of a solution should be short. Next up are costs. These are not so decisive for the development and beginning, as the focus is first on the implementation. However, they will gain in importance with a broad application. In sixth place is adaptability and expandability. In the application considered here, the structure is already defined. In the next steps, only the amount of input data and output data will be extended. However, the data types remain the same. In the second-last place is the accessibility and possibility to use it as a cloud application. This criterion will gain importance with a broad application. At the beginning, however, it plays a subordinate role, as there are only few users. The last place is taken by support. In the long run own know-how should be built up. Figure 3-4 illustrates the determined weighting factors more clearly.

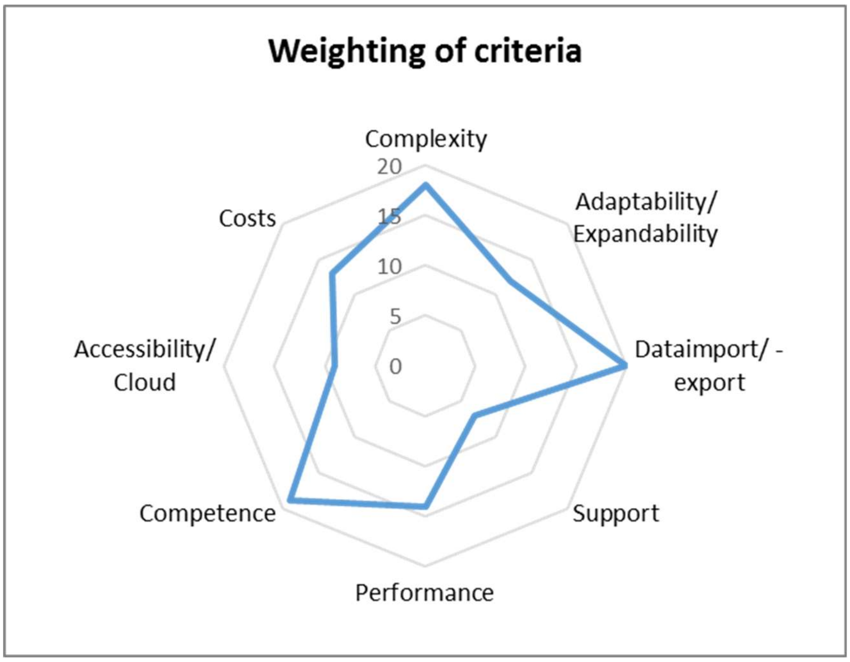


Figure 3-4: Weighting factors of criteria for software selection

Once the weighting factors of the defined criteria have been determined, the potential software is evaluated on the basis of these criteria. For this purpose, five alternatives were selected and briefly analysed. The basic prerequisite for all alternatives is that you can implement ML, including ANNs. They are introduced succinctly below.

PyTorch: This is a Python open source library to develop deep learning applications. Because it is only a library, it offers many possibilities and is very flexible. However, a good knowledge of general programming and specifically of neural networks is required. It supports application-programming interfaces and thus makes it possible to create interfaces to other programs. However, this is unnecessary here, since the economic evaluation is also carried out in Python. Since the computations are carried out on their own hardware, an appropriate one is necessary. The result depends completely on the programming knowledge of the PyTorch user. However, it is free of charge and there is a lot of support in forums, which also requires a certain level of knowledge.

Microsoft Azure ML Studio: It is a tool with which ML solutions can be developed using drag and drop. Due to the drag and drop concept, no programming knowledge is required, which significantly reduces complexity. This makes it less flexible and less expandable. Only knowledge about the ML methods, which one would like to apply, is necessary. The data transfer, data preparation and data processing is easy and intuitive. Standard evaluations of the developed applications are available and quickly show the achieved quality. In addition, it is cloud-based and you only pay for the hardware you use and not for the software itself. It is developed and offered by Microsoft Corporation, which is one of the leading software companies worldwide.

KNIME: It is a free data analysis software. Methods of ML and data mining can be used to develop applications by using a graphical user interface, which makes it less complex than a scripting language. A procedure is implemented by means of a workflow similar to Microsoft Azure ML Studio. It is less flexible than free programming as the methods are only applied. It is available in the Microsoft Azure cloud and the Amazon Web Service cloud.

Amazon SageMaker: This is a cloud-based machine leaning platform. It can be used to develop, train and deploy ML application in the Amazon Web Service cloud. It offers several abstraction levels, so predefined ML models can be used, as well as provided ML algorithms. Experienced users who need a high degree of individualisation can also develop the ML algorithms completely independently. Therefore, it is much more flexible than pure drag and drop applications. However, the interface is not so intuitive, because the elements have to be put together in a program code. This makes it more complex. Since data processing and storage take place in the cloud, only the hardware used is paid for and the functionalities are provided free of charge. This offers scalability in an easy way.

Rapid Miner: It is an environment for creating ML and data mining models. Operators are used in a graphical user interface (GUI) to develop the application. This eliminates almost all code writing. However, the usage is not as intuitive as with Microsoft Azure ML Studio, because it is less clearly arranged. It provides concrete model templates, as well as many different algorithms. One annual licence per user must be purchased. Additionally a server can be added for a fixed price. In the early phase this seems to be much more expensive, because you have to pay fixed defined cost blocks and these are not dependent on usage.

Figure 3-5 shows the benefit analyses of the five alternatives.

	Evaluation (Eva.) Weighted value (W. val.)		PyTorch		Microsoft Azure ML Studio		KNIME		Amazon SageMaker		Rapid Miner	
	Criterion	Weighting	Eva.	W. val.	Eva.	W. val.	Eva.	W. val.	Eva.	W. val.	Eva.	W. val.
1	Complexity	0,16	2	0,32	9	1,45	4	0,64	6	0,96	7	1,13
2	Adaptability/ Expandability	0,11	9	1,45	4	0,64	6	0,96	6	0,96	5	0,80
3	Dataimport/ -export	0,18	4	0,64	8	1,29	6	0,96	4	0,64	8	1,29
4	Support	0,06	3	0,48	8	1,29	5	0,80	9	1,45	6	0,96
5	Performance	0,13	4	0,64	6	0,96	5	0,80	9	1,45	5	0,80
6	Competence	0,17	3	0,48	9	1,45	6	0,96	8	1,29	6	0,96
7	Accessibility/ Cloud	0,08	2	0,32	8	1,29	4	0,64	8	1,29	7	1,13
8	Costs	0,12	10	1,61	7	1,13	10	1,61	8	1,29	3	0,48
	1 = very bad			5,95		9,48		7,39		9,32		7,55
	10 = very good											
		Rank		5.		1.		4.		2.		3.

Figure 3-5: Benefit analysis of possible ML software

PyTorch is in last place, although it is the most versatile and flexible. However, it requires by far the most expertise and is very complex. In the midfield are the less-known software KNIME and the developing environment Rapid Miner. However, the decision as to which to use was made between the cloud platforms Microsoft Azure ML Studio and Amazon SageMaker. They include all required functions, are cloud based, have a pay per use payment model, are not too complex and are developed and offered by experienced companies. This is reflected in the high scores for performance, competence, accessibility/ cloud and costs. Due to Microsoft's reduced complexity and easier data transfer, it ultimately scored the most points. Thus, it was decided to use this as the platform on which the first subprocess of the decision-making process, the ML process to determine the laptop's condition, is developed and executed.

This decision is not universally valid, but was made especially for the decision-making process developed in this work. If the requirements or the structure of the process change or if the application is to be widely used, a new evaluation should be carried out.

3.7 Development of the economic evaluation

This section deals with the development of the economic and ecological evaluation. In the first step, the process of the economic evaluation is defined. This forms the basis for all classes in the software program as well as for the ecological evaluation. Figure 3-6 shows the flow chart of the decision-making software program with focus on the economic evaluation.

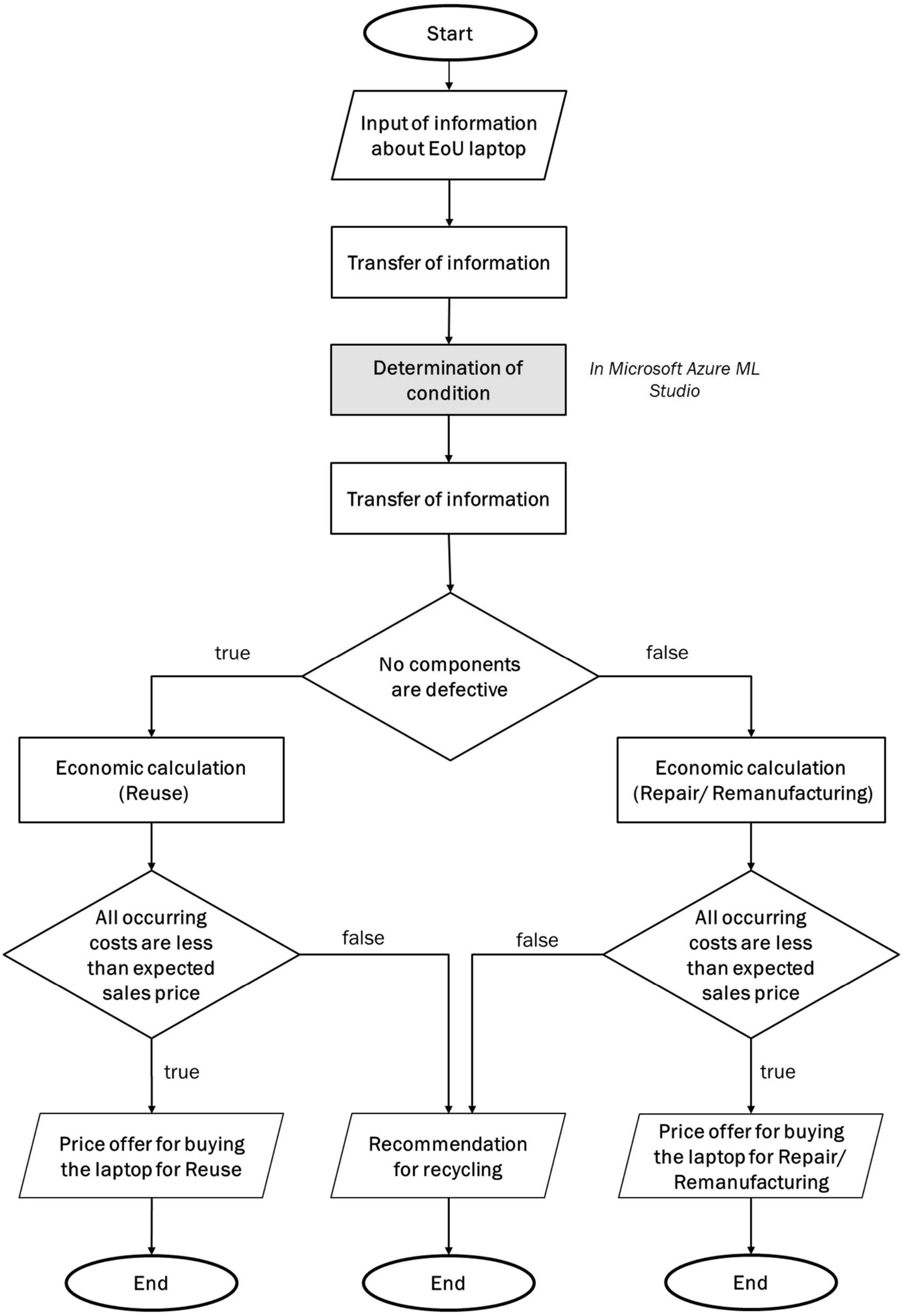


Figure 3-6: Flow chart of the decision-making process

The flow chart of the program was created according to the DIN 66001 and ISO 5807 standard. These standardise the symbols and the general structure in order to create comparability and a common understanding (Hering, 1984).

The program starts when an owner of an EoU laptop, who would like to lead it back into the economic system, enters the required information about the respective laptop via a GUI. The GUI should be implemented on a website as well as in an application. The data is stored in the executive program. After the data has been entered in full, the data is transmitted to Microsoft Azure ML Studio via a communication interface. The respective trained ANN calculates the condition of the component, which is then transmitted as a response to the executive program.

The next step is the economic calculation. The condition of the components as well as the information of the user input are used for this purpose. This is done as a backward price calculation. The possible purchase costs for the laptop should be calculated by the possible selling price and parts of the process costs. Table 3-5 shows the calculation of the possible purchase price.

Table 3-5: Calculation of purchase price

	Calculation of Purchase Price	
1.	Selling price	*Defined by expert or calculated based on historical data*
2.	- Profit	*Selling price * profit margin*
3.	- Risk add-on	*Selling price * risk proportion*
4.	- Capital Commitment costs	*Committed capital * interest rate * average storage time*
5.	- Manufacturing direct costs	*Labour hours * hourly rate*
6.	- Manufacturing overheads	*Manufacturing direct costs * overhead rate manufacturing*
7.	- Material direct costs (spare part)	*Spare part price*
8.	- Material overheads	*Material direct costs * overhead rate material*
9.	= Possible purchase price	

In the first step of the economic calculation, the possible selling price of the EoU laptop under review is determined. This describes the value for which the next laptop in remanufactured condition of this model can be sold. The value is defined either by an expert based on experience or by a mathematical model based on historical data.

Afterwards, the profit and a risk add-on are deducted from the determined sales price. For this purpose, a profit margin and a risk proportion of the selling price are defined. The profit margin describes the share of profit in sales that the company, which uses this system, wants to achieve. The risk add-on serves to compensate for unexpected costs and lost revenues. Additional costs can arise, if the condition of an EoU laptop is estimated to be better than it actually is. This leads to higher repair costs as a result of more spare parts required and more labour hours incurred. Another possibility is that the laptop is not sold for a period of time much longer than the average storage time and therefore causes higher capital commitment costs. Sales may be lower because refurbished laptops cannot be sold for the calculated sales price and the price must therefore be reduced.

Next, the capital commitment costs are deducted. These represent opportunity costs, as the capital tied up cannot otherwise be used profitably. The amount of capital that is tied up is determined by the costs, which will occur for the laptop under consideration until it is refurbished. This includes the purchase costs of the laptop and the spare parts, as well as the labour costs and the overheads for material and production. The calculative interest rate depends on the potential use of the capital for a safe investment. The capital commitment costs are determined as the product of the tied capital, the calculative interest rate and the average storage time, which is based on historical data.

The next position describes the direct manufacturing costs. To calculate these, the determined condition of the laptop is used, Defective components or components with a short remaining service life are replaced. The direct manufacturing costs consist of the labour costs incurred and any machine costs that can be clearly assigned. For calculation the required labour time for a certain task is multiplied by the hourly wage and the machine hour rate by the machine usage time. A distinction is made in hourly wages between repair/ remanufacturing and cleaning for reuse. This should take into account the fact that a higher qualification is required for dismantling, replacing components and assembling than for surface cleaning. In addition, different tools and machines are used. For each laptop model and component, the required durations are defined on the basis of empirical values and stored in a small data base. The sixth position is held by the manufacturing overhead costs. This describes the manufacturing costs, which cannot be assigned clearly to one cost object. These include energy costs, calculated depreciation on machines and tools, and costs for auxiliary materials. These are calculated on the basis of the overhead rate of the manufacturing direct costs. Since the overheads cannot be determined precisely beforehand, the rate of overhead is defined on the basis of experience.

Finally, the material direct costs and the material overhead costs are allocated. This is done in a similar way to the manufacturing costs based on the determined condition of the laptop. For each laptop model, the installed components are stored. If a component has to be replaced, the purchase costs of the spare part are charged as direct material costs. Material costs that cannot be clearly allocated are added as overhead costs by means of an overhead rate.

No transport costs are deducted. The owner of the EoU laptops is responsible for transport of the laptop to the determined institution. Furthermore, the new buyer of the reconditioned laptop has to pay for the delivery from the institution to the desired address.

If the calculated purchase price is above zero, it will be offered to the laptop owner. If the calculated value is less than zero, the laptop owner is recommended to recycle it. . The owner of the laptops is offered a price for the resources used in the laptop. This price depends on the new price of the laptop model.

3.8 Development of ecological evaluation

In addition to the previously developed economic evaluation of the EoU laptop, an ecological evaluation of the laptop is also to be carried out. To this end, it must first be defined on the basis of which parameters this evaluation is conducted. Prakash et al. (2016) and O'Connell and Stutz (2010) use the Product Carbon Footprint (PCF) in order to describe the life cycle greenhouse emissions, which are released as part of the processes of creating, transporting, storing, using, providing, recycling and disposing of products. These are responsible for the greenhouse effect, which leads to global warming. The unit of measurement used is ‘kilogram CO2 equivalents’. This makes it possible to summate the environmental impact of the different green house gases (GHG) and thus create comparability with other products or processes. The PCF can be divided into several blocks, which are oriented to the process causing them. Figure 3-7 illustrates a bar graph of the different components of the PCF of an average laptop.

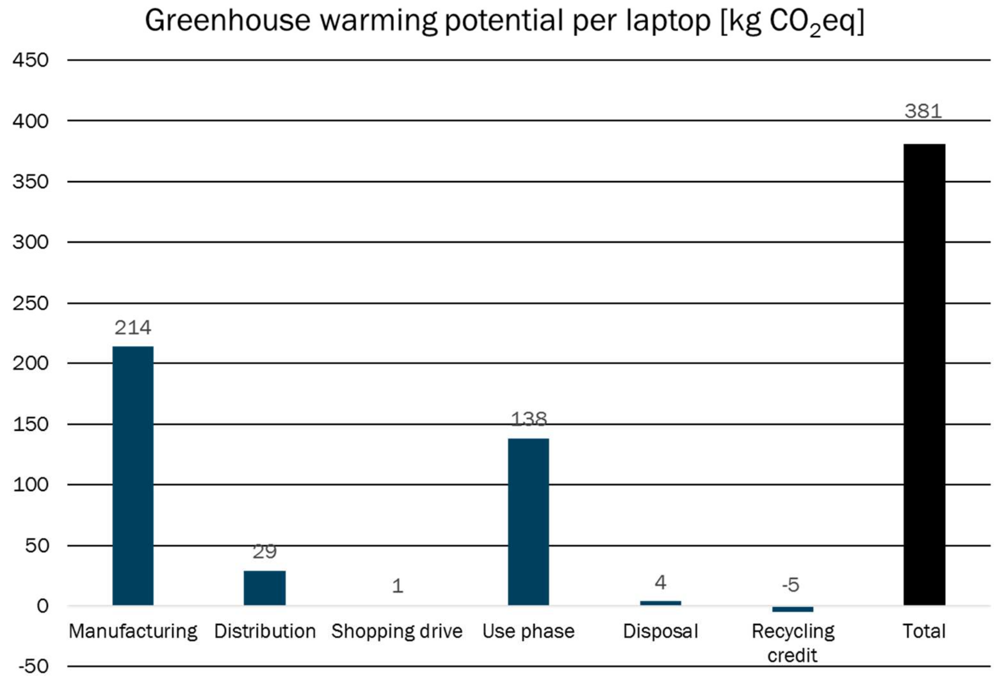


Figure 3-7: Greenhouse warming potential per laptop. Own illustration based on Prakash et al. (2012, p. 26)

It turns out that there are two main factors, the manufacturing and the use phase. Under the label of manufacturing, emissions are recorded that are generated during the extraction of resources, production of components, transport between the different production sites and assembly. Distribution describes the transport of the finished laptop from the manufacturer to the point of sale. The shopping drive is the customer's journey to the location to buy the laptop and back. In the usage phase, the emissions generated for the generation of the required energy are recorded. The disposal label records the emissions of transports and processes to dispose of the laptop. Finally, the amount of recycling credit is deducted. This describes the difference in emissions saved by recycling and the resulting use of secondary resources instead of primary resources. This is due to the fact that primary extraction of precious metals is associated with considerably higher environmental impacts than the secondary extraction (Prakash et al., 2012, p. 5).

In this work, the information about recycling credit is communicated to the owner of the EoU laptop if the previous economic evaluation comes to the conclusion that the reuse or remanufacturing of the laptop is not profitable. This should motivate the owner to take the laptop to a collection point in addition to the offered price for the laptop to recycle it. Furthermore, the information for manufacturing and disposal of the laptop is used for the ecological evaluation. These are the parts of emission, which can be prevented, by the reuse or remanufacturing of an EoU laptop. Figure 3-8 illustrates three different scenarios of laptop reuse/ remanufacturing.

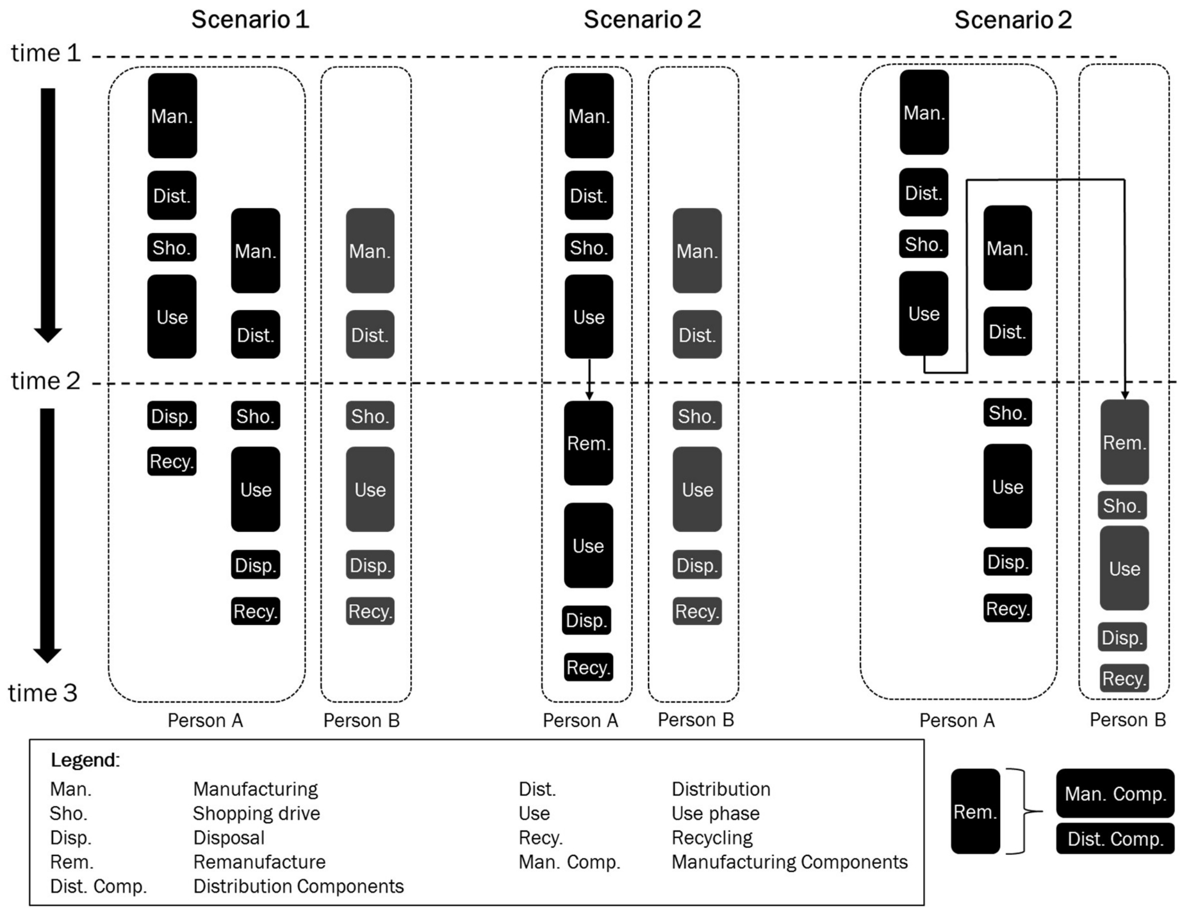


Figure 3-8: Scenarios of laptop reuse/remanufacturing

In the different scenarios, it is assumed that person A buys a laptop, which loses its functionality at time 2. At time 2, person A and person B should have a functioning laptop.

In scenario 1, person A disposes of the first laptop after its useful life and buys a new one. Person B also buys a new laptop. In total, 3 times the whole emissions of the PCF are produced.

In scenario 2, person A decides to have the laptop remanufactured and reuse it afterwards. Person B buys a new laptop at time 2. In this scenario, emissions from manufacturing, shopping tour and disposal, as well as the recycling credit only occur twice. However, this will result in emissions from the manufacture and transport of the spare parts.

In scenario 3, person A returns the EoU laptop and buys a new one. The first laptop of person A is remanufactured and is then purchased by person B. The large emission parts are analogous to scenario 2.

It can be seen that in scenario 2 and 3 significantly fewer emissions are produces, this also applies to the use of resources.

As explained, the reuse of a laptop without replacement of any parts would prevent the generation of the emissions for the complete manufacturing phase and the disposal for a new laptop. This amount is reduced by the emissions generated during the production and transport of the required spare parts in case that components have to be replaced. The following equation is therefore obtained for calculating the emissions saved:

$$saved\ emissions = tot \quad emission \quad \sum_{i=1}^{n} emission\ of\ spare\ part \tag{3.1}$$

The ANN identifies the required spare parts. This information is evaluated with the emissions caused per spare part type. In order to calculate the ecological benefit of the recovery, the emissions per component are roughly determined. The proportion of the individual components in the total emissions of the manufacturing phase of a laptop were derived on the basis of O'Connell and Stutz (2010).

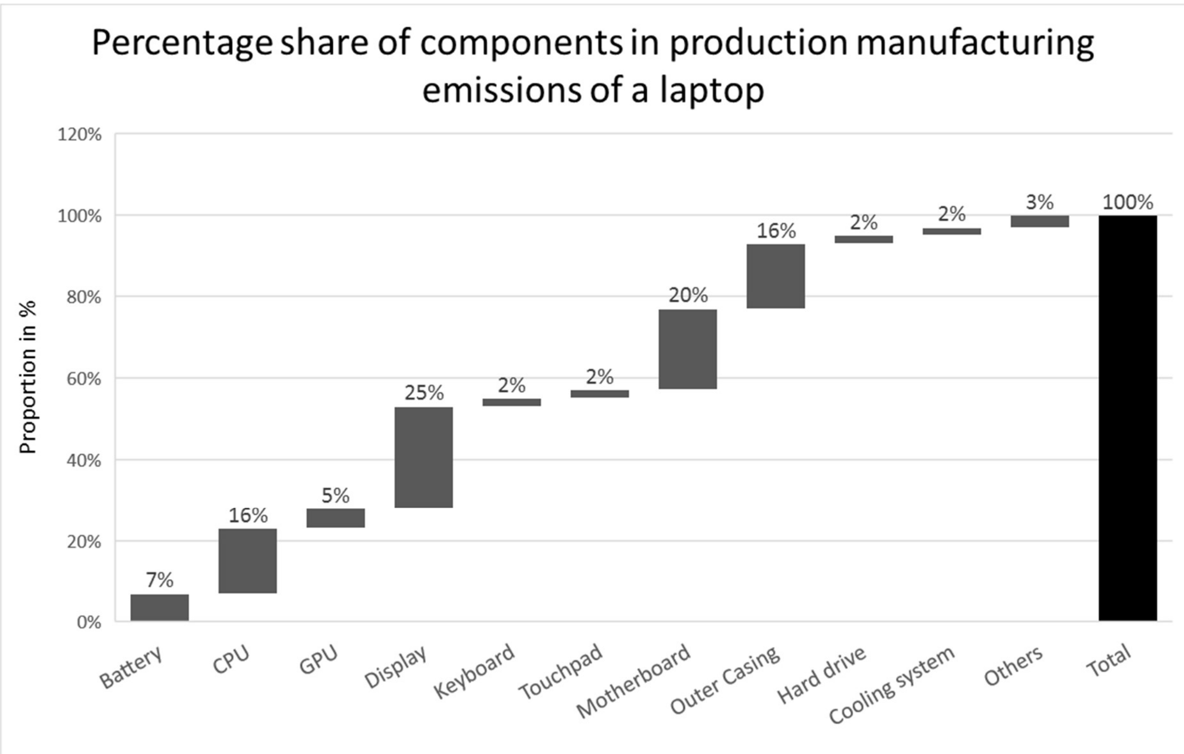


Figure 3-9: Proportion of emissions of a laptop caused per component. Own illustration based on O'Connell and Stutz (2010)

The values for the battery, display, keyboard, outer casing and hard drive are calculated directly based on the absolute values of the components mentioned in relation to the total amount of emissions. The listed value of the motherboard is mainly divided into the shares of the motherboard itself, the CPU and touchpad. In addition, other components are attributed shares, as all connection parts were allocated to the motherboard. The values for the touchpad and cooling system are estimated based on other components. This distribution of the emissions to the components has no claim to correctness. It serves to show the procedure and implementation in the whole system. In practical use, emission values per component would have to be determined more precisely. In addition, different designs of components could be distinguished. This would be simplified if companies were obliged to make a PCF publicly available for the components of their products. Nevertheless, such a rough estimate already gives the customer an impression of the impacts of the decision. The emissions for the distribution of a spare part are calculated based on the value of the laptop's distribution. Each of the 10 spare part components is assigned 10% of the laptop's distribution value. This is for each component 2.9 kg CO_2eq (=29 kg CO_2eq / 10).

Finally, the following equation results from the calculation of the emissions per spare part, which is used in (3.1) to calculate the sum of emissions of all required spare parts:

$$emissions\ of\ spare\ part = manufacturing\ emissions + 2.9\ kg\ CO_2eq \tag{3.2}$$

This is calculated for an EoU laptop with a defective battery (7% of total emissions) and hard drive (2% of total emissions) as an example:

$$emissions\ spare\ battery = (217 * 0.07) kg\ CO_2eq + 2.9\ kg\ CO_2eq = 18.09\ kg\ CO_2eq$$

$$emissions\ spare\ hard\ drive = (217 * 0.02) kg\ CO_2eq + 2.9\ kg\ CO_2eq = 7.24\ kg\ CO_2eq$$

$$saved\ emissions = 217\ kg\ CO_2eq - (18.09 + 7.24) kg\ CO_2eq = 191.64\ kg\ CO_2eq$$

Scenario 2 and 3 therefore save 191.64 kg CO_2eq compared to Scenario 1.

3.9 Chapter summary

The purpose of this chapter is the theoretical development of the decision-making process. The requirements are derived from the problem analysis and a structure of the process is developed. It consists of two parts, which are the determination of the condition of the laptop on component level and the economic and ecological evaluation. For the first part ANNs are selected as ML method and Microsoft Azure ML Studio as software. The structure and the process of the evaluations are defined on the basis of the output of the ANNs. The developments of this chapter serve the practical implementation in further chapters.

Chapter 4
Simulation of the data

This chapter describes the simulation of datasets from EoU laptops. These are intended to represent the data that can later be captured in reality to train the ANNs. Therefore, correlations are implemented in the data to reflect the relationships. At the beginning the values the individual can assume are defined. Then the independent variables are simulated, which are used for the calculation of the dependent variables.

4.1 Definition of the classes of the variables

A single dataset consists of 22 variables. Of these 12 are descriptive variables that serve as input for the neural network. They are also called 'independent variables'. The remaining ten variables form the result of the ANN. These are the variables to be described or also called 'dependent variables'. They are the solution to the input. After a training phase, ANN should independently determine the output to an input. Figure 4-1 shows the independent variables that describe the laptop model and have some information about the usage of the laptop. Furthermore, it illustrates the possible values of the variables.

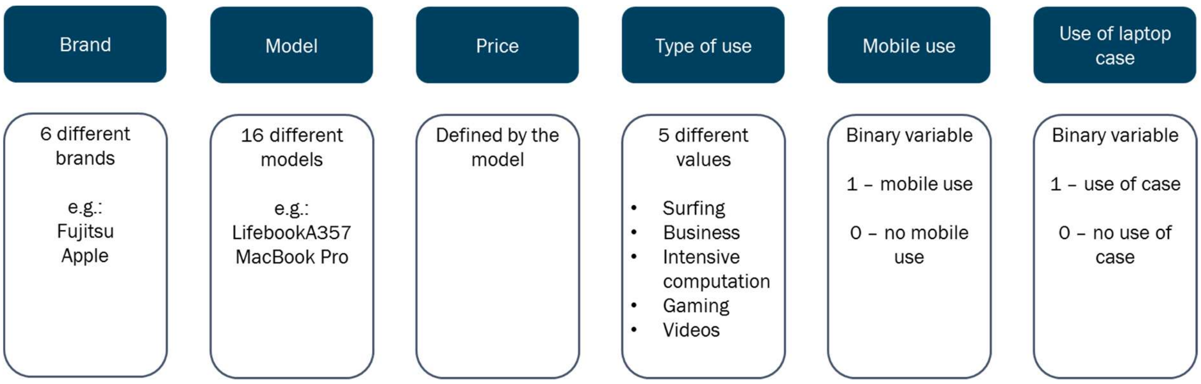


Figure 4-1: Independent variables concerning the laptop model and usage

For simulation, 16 laptop models of six brands are simulated. The generated data are freely simulated and do not represent real data. The usage type assumes one of five predefined values. 'Surfing' describes the use as a private laptop, with which online shopping and other everyday tasks such as online banking are carried out. Laptops that are used for business purposes and are often used for applications such as PowerPoint, Excel or enterprise resource planning systems are assigned the value ‘business’. The third type of use describes ‘intensive computation’ use. This includes laptops on which simulations, data processing or graphical editing are carried out. ‘Gaming’ laptops are also used for computing intensive tasks. The category ‘videos’ includes laptops which are used in a similar way to surfing laptops. However, these are primarily used for the display of movies and videos. The variables ‘mobile use’ and ‘use of laptop case’ can only have the values 1 or 0, which corresponds to a yes or no.

Figure 4-2 shows the independent variables, which describe the external condition of the laptop. They describe obvious external damages of different parts of the laptop that the application user can detect without disassembling the laptop. These variables can have between three and five different values. However, only one value per variable can be assigned to a laptop. If a component has several damages, its variable is assigned the value of the largest damage. In Figure 4-2 these are arranged from top to bottom.

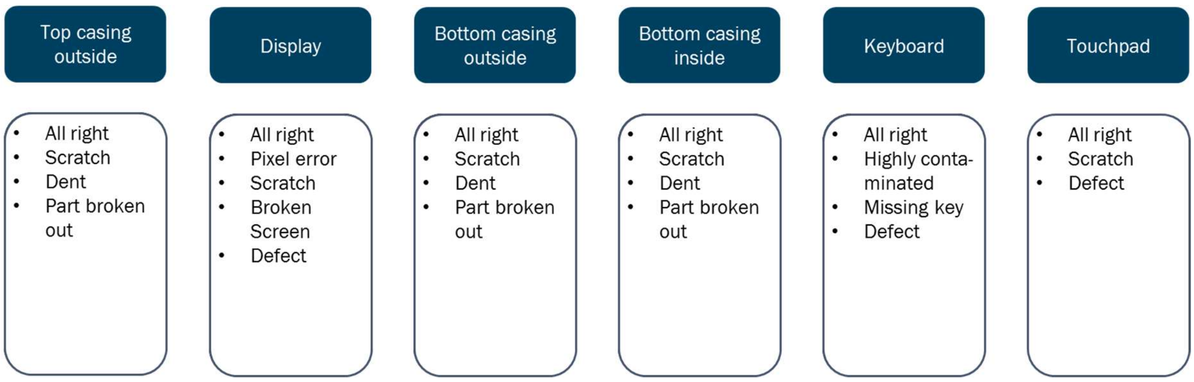


Figure 4-2: Independent variables concerning external damages of the laptop

4.2 Simulation of independent variables

The simulation is performed with RStudio, because this software provides many functions for data simulation. Furthermore, data can easily be graphically displayed and an export as a comma separated text (CSV) file is possible. This is necessary in order to import the data to Microsoft Azure ML Studio. Figure 4-3 shows the procedure for how the data is simulated.

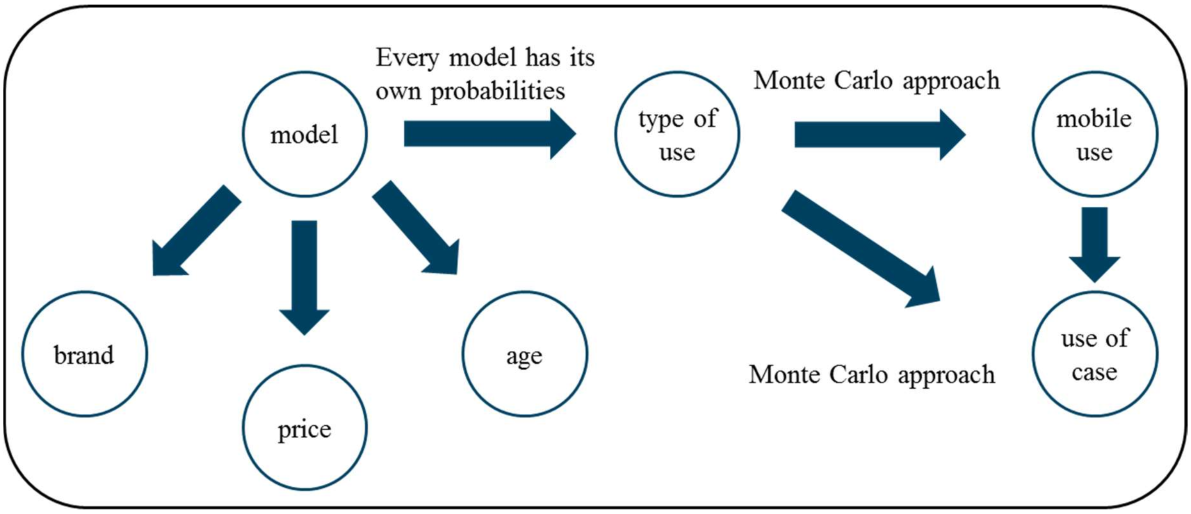


Figure 4-3: Procedure of data simulation

In the first step 250,000 laptops are created by choosing a model randomly out of 16 possible models. These models are listed in Table 4-1. Next, the values of the variable brand are determined, which is clearly defined by the model. Each model is also clearly assigned a price.

Table 4-1: Laptop models, brands and price

Laptop model	Brand	Price [€]
Fujitsu Lifebook A357	Fujitsu	479
Fujitsu Lifebook U728	Fujitsu	1399
Fujitsu Lifebook U729	Fujitsu	1299
MacBook Pro	Apple	1999
Mac Book Air	Apple	1099
HP EliteBook 830 G6	HP	1333
HP EliteBook 850 G6	HP	1466
HP EliteBook 1040 G4	HP	1796
HP EliteBook 1050 G1	HP	3323
Dell Inspirion 13 5000	Dell	519
Dell Precision 5520	Dell	1906
Lenovo Thinkpad T580	Lenovo	1724
Lenovo Thinkpad E490	Lenovo	1011
Acer Swift 1	Acer	599
Acer Aspire 5	Acer	999
Acer Aspire 7	Acer	1299

The next step is assigning an age to each dataset of a laptop. The age is given in months. A normal distribution with an average of 36 months and a standard deviation of eight months was used as probability distribution. Figure 4-4 shows the histogram of the age. The axis of ordinates represents the frequency and the axis of abscissas represents the ages. It shows the typical bell shape of the normal distribution around the value of 36 months.

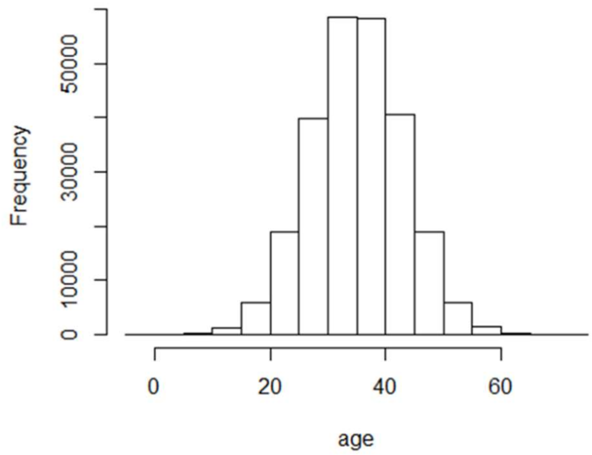


Figure 4-4: Histogram of age

In the next step, the usage types are generated. This is done by using defined probabilities. For this purpose, probabilities of the five different usage types are defined for each model. This is based on the brand and price as well as the hardware configuration. Low-priced laptops tend to be used as surfing or video laptops. Mid segment laptops (around 1000 – 1500 €) are mostly used as business laptops. On the other hand, expensive laptops with good hardware in terms of processor, CPU and GPU are more often used for computational intensive tasks and gaming. Table 4-2 lists the probabilities for the different usage types for each laptop model. This should also represent that laptops in general are used less for computational intensive tasks and gaming, but more for business or simple private tasks.

Table 4-2: Probabilities of type of usage for every model

Laptop model	**Prob. of "surfing"**	**Prob. of "business"**	**Prob. of "comput. intensive"**	**Prob. of "gaming"**	**Prob. of "videos"**
Fujitsu Lifebook A357	25 %	30 %	15 %	5 %	25 %
Fujitsu Lifebook U728	8 %	66 %	18 %	2 %	16 %
Fujitsu Lifebook U729	8 %	64 %	20 %	2 %	16 %
MacBook Pro	24 %	40 %	15 %	1 %	26 %
Mac Book Air	30 %	32 %	10 %	1 %	33 %
HP EliteBook 830 G6	10 %	58 %	18 %	4 %	10 %
HP EliteBook 850 G6	18 %	60 %	18 %	3 %	11 %
HP EliteBook 1040 G4	12 %	59 %	21 %	5 %	3 %
HP EliteBook 1050 G1	3 %	30 %	50 %	15 %	2 %
Dell Inspirion 13 5000	30 %	30 %	10 %	2 %	28 %
Dell Precision 5520	7 %	65 %	24 %	2 %	12 %
Lenovo Thinkpad T580	10 %	63 %	20 %	2 %	16 %
Lenovo Thinkpad E490	29 %	41 %	16 %	2 %	24 %
Acer Swift 1	46 %	10 %	6 %	1 %	42 %
Acer Aspire 5	20 %	40 %	7 %	3 %	20 %
Acer Aspire 7	16 %	42 %	10 %	3 %	29 %

After the usage type has been determined, the next step determines the variable 'mobile use'. This indicates whether or not the laptop was frequently used for mobile purposes. For this the so-called Monte Carlo approach is used. The Monte Carlo approach is based on the use of random numbers (Fahrmeir, Heumann, Künstler, Pigeot, & Tutz, 2016, p. 298; Klein, 2010; Schmallowsky & Reimers, 2015, p. 5). It first defines a probability that a certain laptop was in mobile use. This probability depends on the type of use. Table 4-3 shows the probabilities that a laptop was in mobile use for the individual types of use. A random number in the interval [0, 1] is then generated for each dataset. This is compared with the corresponding probability. If the random number is smaller than the probability, the variable 'mobile use' assumes the value 1, so the laptop was in mobile use. If the random number is higher than the probability, the variable 'mobile use' assumes the value 0, so the laptop was not used for mobile use.

Table 4-3: Probability of mobile use depending on type of use

Type of use	**Probability of mobile use**
Surfing	10 %
Business	40 %
Computational intensive	18 %
Gaming	4 %
Videos	12 %

Next, the variable 'use of case' is simulated. The Monte Carlo approach is used in the same way as for the variable 'mobile use'. The probabilities are based on the type of use and the mobile use. The probabilities are calculated using an additive formula. For each type of use, a probability is defined that a laptop case was used. If the laptop was in mobile use, 50 % is added to this probability. Table 4-4 shows these probabilities depending on the type of use and whether it was in mobile use.

Table 4-4: Probabilities of use of case depending on type of use and mobile use

Type of use	**Probability of use of case**
Surfing	15 %
Business	35 %
Computational intensive	25 %
Gaming	2 %
Videos	10 %
Mobile use	
Yes	50 %
No	0 %

Now the last descriptive variables are simulated. These describe the external damages to the six defined components, which are the top casing outside, display, bottom casing outside, bottom casing inside, keyboard and touchpad. Different damages are defined for the different components. Each of these damages is assigned a probability, which indicates how likely it is that an EoU laptop has the corresponding damage. In this simulation, the probabilities for the damages are independent from other variables, which does not correspond to reality. Table 4-5 shows the different damages and the corresponding probability.

Table 4-5: Probabilities of external damages

Component	**Categories and probabilities**				
Cover outside	All right	Scratch	Dent	Part broken out	
	85 %	8 %	5 %	2 %	
Display	All right	Defect	Pixel error	Scratch	Broken screen
	80 %	5 %	7 %	3 %	3 %
Base outside	All right	Scratch	Dent	Part broken out	
	85 %	8 %	5 %	2 %	
Base inside	All right	Scratch	Dent	Part broken out	
	87 %	10 %	2 %	1 %	
Keyboard	All right	Highly contaminated	Missing key	Defect	
	79 %	15 %	2 %	4 %	
Touchpad	All right	Scratch	Defect		
	85 %	12 %	3 %		

Table 4-6 shows by way of example the descriptive variables of ten datasets out of the 250,000. These form the input for the artificial neural network. The condition of the ten previously defined components are to be determined on the basis of the values of these variables.

Table 4-6: Independent variables of simulated data

model	age	typeOfUse	mobile	Case	price	brand	coverOut	display	baseOutside	baseInside	keyboard	mousepad
Dell Precision 5520	32	gaming	0	0	1906	Dell	All right	All right	All right	All right	Highly contaminated	All right
Lenovo Thinkpad E490	26	computationally int	0	0	1011	Lenovo	All right	Defect	All right	Scratch	Highly contaminated	All right
MacBook Air	36	surfing	0	1	1099	Apple	All right	All right	All right	All right	All right	Scratch
HP EliteBook 830 G6	41	normal business	0	1	1333	HP	All right	All right	All right	All right	All right	All right
Fujitsu Lifebook U729	29	normal business	0	0	1299	Fujitsu	All right	All right	All right	Scratch	All right	All right
HP EliteBook 830 G6	32	normal business	1	0	1333	HP	All right	PixelErrors	All right	All right	All right	All right
HP EliteBook 1040 G4	32	videos	0	1	1796	HP	Scratch	PixelErrors	All right	All right	Highly contaminated	All right
Acer Swift 1	41	surfing	0	0	599	Acer	All right	PixelErrors	All right	All right	All right	All right
Fujitsu Lifebook U729	35	normal business	1	1	1299	Fujitsu	All right	All right	BrokenOutPart	All right	All right	All right

4.3 Simulation of dependent variables

After the descriptive variables have been generated, the dependent variables are simulated, which are to form the output of the ANN. For this purpose, the Monte Carlo approach is again used. This should implement the so-called noise in the data. The noise is used to represent the circumstances of reality, which cannot be reproduced by descriptive models. If it does not occur, enough complex models could accurately predict each outcome.

As mentioned before, the independent variables simulated in Section 4.2 are used to determine the condition of the following laptop components, which make up the dependent variables:

1. Battery
2. CPU
3. GPU
4. Display
5. Keyboard
6. Touchpad
7. Motherboard
8. Housing
9. Hard drive
10. Cooling system

For the Monte Carlo approach, probabilities must again be determined as to whether the components are in good condition or whether they need to be replaced. For the dependent variables, these cannot simply be defined or determined by using a simple additive method, as there are too many combinations. Therefore, the probabilities are calculated with the logistic function, which was elaborated in chapter 2.3.3. This guarantees that the determined values are in the range 0 to 1. This would not be guaranteed when using the linear function. To do this, the betas which are used in the equation of the logistic function (see equation 2.2), must first be determined. They describe how strongly the respective independent variable influences the dependent variable. The values an independent variable can accept must be taken into account. The betas must be determined separately for each laptop component, since they react differently to the different influences. For example, it is implemented that the condition of the battery depends more on the age of the laptop than on the use of a protective case. In order to be able to process the categorical variables, so-called dummy variables must be created. Each dummy variable presents a possible value of a categorical variable and indicates whether the categorical variable has this value or not. Equation 3.1 and 3.2 represent the formulas for calculating the probability that the battery needs to be replaced. They show how the betas are used.

$$p(X) = \frac{e^z}{1 + e^z} \tag{3.1}$$

$$\begin{aligned} z = {} & -9.2 + 0.18 * age + 3 * mobileUse + (-1.0) * useOfCase \\ & + (-0.0011) * price \\ & + 6.0 * typeOfUseCompIntensive \\ & + 5.0 * typeOfUseGaming \\ & + 1.1 * caseOutsideBrokenOtPart \\ & + 1.2 * displayBrokenScreen \\ & + 0.7 * baseOutsideDent \\ & + 1.5 * baseOutsideBrokenOutPart \\ & + 0.7 * baseInsideDent \\ & + 1.5 * baseInsideBrokenOutPart \\ & + 0.4 * touchpadDefect \end{aligned} \tag{3.2}$$

The intercept can be used to move the calculated probabilities in a certain direction. This shifts the centre of the probability distribution. This method is used when the absolute proportion of defective batteries is too high or too low. The betas for the models and brands have been set to zero and therefore have no influence on the probability. Since the price has a very strong correlation to the laptop model (the price is defined by the model), the models are already taken into account. The influence of the brands is not taken into account in order not to depict one brand worse than others. However, it would have been possible to implement connections here, such as owners of a certain brand being more careful with their laptops and therefore less often cause external damage. Therefore, it can also be useful to collect data to which one does not suspect any connection at the beginning. However, ANNs are able to uncover such connections.

For the further determination of the betas for the calculation, the stressing factors developed in Sections 3.3 and 3.4 are used. Table 4-7 shows that the age, mobile use, computational intensive tasks and gaming have a strong positive effect on the probability that the battery will have to be replaced. The battery is a wearing part that ages. In addition, it is temperature-sensitive, which is influenced by the computing intensity, which is described by the usage type. Furthermore, the temperature depends on the ability to dissipate heat, which depends on the amount of dust and dirt in the ventilation, which tends to be more if the laptop is in mobile use. In addition, the probability increases with external damages like dents or broken out parts, which indicate a fall. Both the use of a protective case and a high price reduce this probability. This is because high-quality batteries lose capacity more slowly and a case protects against dust.

Table 4-7: Factors of independent variables to determine dependent variables

Variable	Dummy	Battery	CPU	GPU	Display	Keyboard
Intercept		-9.2	-6.5	-6.5	-8.1	-8.1
Age		0.18	0.10	0.08	0.10	0.08
Mobile use		3	1.5	1.0	4.0	1.7
Use of case		-1.0	-2.0	-1.0	-2.5	-1.4
Price		-0.0011	-0.0014	-0.0014	-0.0014	-0.0018
Model	All models	0	0	0	0	0
Type of use	Surfing	0	0	0	0	0
	Business	0	0	0	1.5	2.5
	Comp. intensive	6	6	3.5	1.0	0.7
	Gaming	5	4.3	6.5	3.5	3.5
	Videos	0	2	3	2.5	0
Brand	All brands	0	0	0	0	0
Cover Outside	All right	0	0	0	0	0
	Scratch	0	0	0	1.0	0.2
	Dent	0	0.4	0.7	2.3	0.4
	Part broken out	1.1	1.0	1.5	7.0	0.7
Display	All right	0	0	0	0	0
	Defect	0	0.4	5.8	7.0	0.2
	Scratch	0	0	0.5	6.0	0.4
	Broken screen	1.2	0.8	1.5	20.0	0.8
	Pixel errors	0	0	2.2	15.2	0.2
Base outside	All right	0	0	0	0	0
	Scratch	0	0	0	0	0.2
	Dent	0.7	0.8	0.7	1.1	1.3
	Part broken out	1.5	1.7	1.5	3.5	2.8
Base inside	All right	0	0	0	0	0
	Scratch	0	0	0	0.8	1.0
	Dent	0.7	0.6	0.6	2.4	1.9
	Part broken out	1.5	1.4	1.4	5.4	4.4
Keyboard	All right	0	0	0	0	0
	Missing key	0	0.2	0.1	0.6	14
	Contaminated	0	0.5	0.6	0.9	7.5
	Defect	0	0	0.5	0.8	17
Mousepad	All right	0	0	0	0	0
	Scratch	0	0.1	0.2	0.3	1.2
	Defect	0.4	0.5	0.4	0.6	2

The betas for determining the probabilities that the CPU or the GPU have to be replaced are very similar. The CPU and the GPU produce heat and are exposed to it. Therefore, age, mobile use and computational intensive tasks have a strong positive impact. External damages also increase the probability, as connection to the components can be damaged through a fall. The price and use of a case have a negative influence, similar to the battery. Defects in the display can also be caused by a defective GPU.

The condition of the display is mainly determined by mobile use and external damage. The display is most frequently damaged by falls, which tend to occur during mobile use. In addition, graphic representation like gaming and videos have an influence on the probability that the display has to be changed. Furthermore, the physical condition of the display is easily apparent and is one of the descriptive variables.

The decisive variable for the condition of the keyboard is the obvious external condition of the keyboard, since this is visible without disassembly and is recorded as an own independent variable. In addition, it is strongly influenced by the type of use. Input-intensive forms like business use or gaming are to be emphasised here. Mobile use makes the keyboard dirty faster and a case protects it.

Table 4-8 shows the second part of the betas for determining the dependent variables. The condition of the touchpad is determined similarly to that of the keyboard. However, here the independent variable of the external condition of the touchpad is decisive and gaming does not stress the touchpad as much, because gamers often use an external device. In addition, surfing and watching videos have an influence, since few input commands can be done well with the touchpad.

The motherboard is again a heat-sensitive component, which is why it increases the probability of it needing to be replaced through computational intensive and mobile use. It also has some physical links so it can easily be damaged by falls, which can be noticed by external damage.

The housing can primarily be damaged by mechanical influences. Therefore, mobile use and the use of a protective case have a great influence on its condition. In addition, the condition of the housing is already described by the six independent variables that determine the external damage.

Due to its design, the hard drive is also susceptible to falls. It is also more frequently used for business and computational intensive tasks. However, this is changing due to the growing use of the cloud.

The cooling system is mainly exposed to dust and heat. That is why the betas are the largest for mobile use and tasks, which generate a lot of heat. A case protects against dust and therefore has a negative beta. A dirty keyboard can indicate that the laptop has been exposed to a lot of dirt and dust and has not been maintained, which has also a strong positive effect on the probability that the cooling system needs to be replaced. A high cost price often leads to a more careful handling of the laptop and therefore has a negative influence on the probability besides the quality aspects. This also applies to other components.

Table 4-8: Factors of independent variables to determine dependent variables

Variable	Dummy	Mousepad	Motherboard	Housing	Hard drive	Cooling system
Intercept		-7.5	-7.8	-7.8	-7.6	-8.3
Age		0.08	0.12	0.10	0.13	0.15
Mobile use		2.5	3.9	4.5	2.0	5.2
Use of case		-1.4	-2.2	-3.0	-1.0	-3.0
Price		-0.0018	-0.002	-0.002	-0.0022	-0.0021
Model	All models	0	0	0	0	0
Type of use	Surfing	2.0	0.5	1.2	0	2.0
	Business	1.7	1.5	1.7	2.8	3.4
	Comp. intensive	0.7	3.8	0.3	3.4	2.7
	Gaming	1.0	2.8	0	0.8	1.8
	Videos	1.8	1.2	0.9	1.5	2.2
Brand	All brands	0	0	0	0	0
Cover Outside	All right	0	0	0	0	0
	Scratch	0.2	0.2	0.5	0.1	0.1
	Dent	0.8	0.3	4.6	0.8	0.2
	Part broken out	1.3	1.8	12.8	1.8	0.5
Display	All right	0	0	0	0	0
	Defect	0.2	3.8	1.4	1.0	0.6
	Scratch	0.4	0.1	0.4	0.1	0.1
	Broken screen	0.8	1.5	1.9	0.9	0.7
	Pixel errors	0.2	1.3	0.8	0.7	0.4
Base outside	All right	0	0	0	0	0
	Scratch	0.2	0.3	0.4	0.2	0.1
	Dent	1.3	1.5	4.5	1.5	0.3
	Part broken out	2.8	4.8	11.8	1.8	0.5
Base inside	All right	0	0	0	0	0
	Scratch	1.0	0.4	0.9	0.1	0.1
	Dent	1.9	1.8	5.8	1.6	0.4
	Part broken out	4.4	4.3	17.3	1.9	0.6
Keyboard	All right	0	0	0	0	0
	Missing key	2.0	0.8	1.4	0.6	0.4
	Contaminated	3.0	1.9	1.9	0.9	2.2
	Defect	4.0	1.5	1.5	0.5	1.5
Mousepad	All right	0	0	0	0	0
	Scratch	3.2	0.1	0.8	0.2	0.1
	Defect	17	1.7	1.5	0.7	0.4

For each simulated object and component a probability is calculated, that the respective component has to be exchanged, by using the defined betas and the logistic function. This probability is then compared with a generated random number. If the random number is smaller than the calculated probability, the component is regarded as defective. If the random number is larger, the component does not have to be replaced. This process must be performed for all 250,000 objects and for each component. Table 4-9 represents the values of the dependent variables of 10 exemplary datasets. All simulated data is exported completely as a text file to be able to import them into Microsoft Azure ML Studio.

Table 4-9: Dependent variables of simulated data

battery	cpu	gpu	display	keyboard	mousepad	motherboard	housing	hardDrive	coolingSystem
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0

The entire source code of the simulation of the data can be found in the Appendix A.

4.4 Chapter summary

This chapter serves to simulate the data. In the first step, the values that the variables can assume are defined. In the case of categorical variables, these are different classes. Subsequently, the independent variables are simulated on the basis of probabilities. These are used in the Monte Carlo approach to simulate the values of the dependent variables, which describe the conditions of the laptop's components.

Chapter 5
Implementation of the decision-making process

This chapter deals with the implementation of the ANNs in Microsoft Azure ML Studio and the implementation of the economic and ecological evaluation in Python.

5.1 Implementation of the artificial neural networks

At the beginning, a new experiment is created in Microsoft Azure ML Studio. For each component, for which the condition is to be determined, a separate ANN must be created and trained. However, this process is nearly identical for each component. Figure 5-1 represents this process for one ANN.

As Section 3.6 explains, the main function blocks are inserted via drag and drop in Microsoft Azure ML Studio. The first block describes the data import. Here it is necessary to select the desired data, which must already have been uploaded to the cloud beforehand.

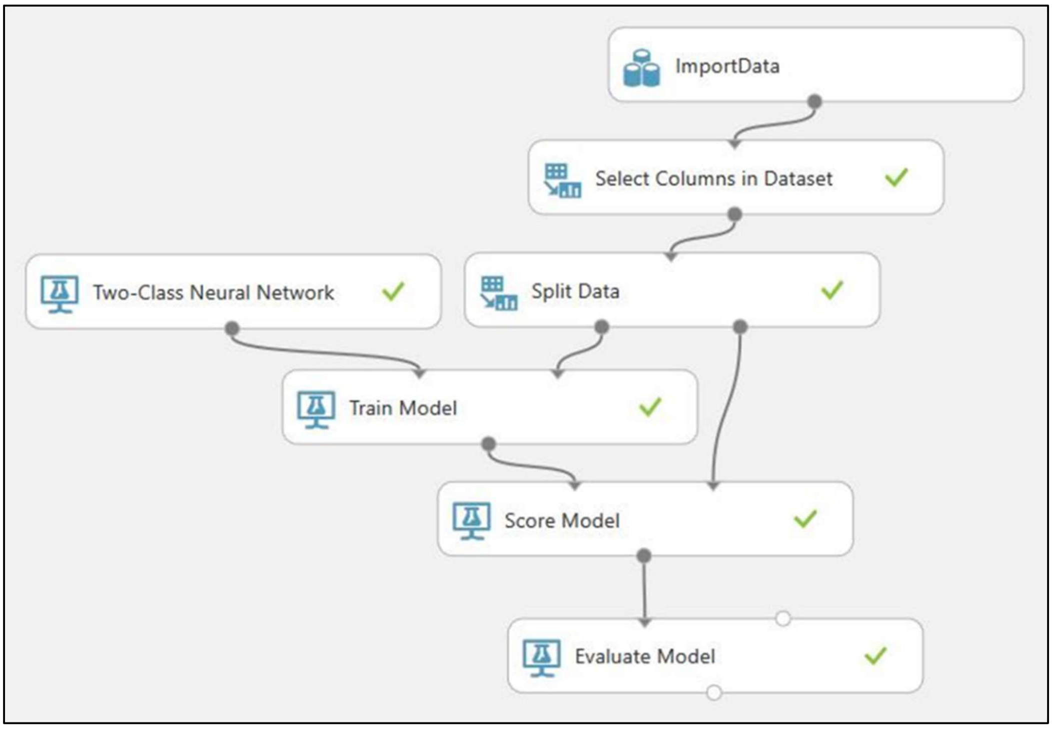


Figure 5-1: Process of generating and training the artificial neural network

The next step is to select the columns of the dataset, which should be used to train the network. For this purpose, all independent variables and the dependent variable, which the ANN is to determine, are selected. The first ANN is trained for the variable "conditionBattery", for example.

Afterwards the 250,000 datasets are divided into training and test data. Literature recommends a division into 80 % training data and 20 % test data (Backhaus et al., 2015, p. 322). However, a lot of data is available here and therefore the training data share is reduced slightly to 70 %, which still represents 175,000 training datasets. The division is random, but a seed is set that the same division can be reproduced when the program is repeated from the beginning.

Furthermore, the desired ML method is now selected. As elaborated in Section 3.5, the two-class feedforward perceptron is selected. The hidden layer specification determines the structure of the hidden layer. The fully-connected case is chosen. In this case there is only one hidden layer. This should be sufficient for the complexity of the regarding data and therefore reduces the amount of required training data and processing time. Furthermore, overfitting is less likely to occur. Each neuron of this hidden layer is connected to each input neuron and the output neuron.

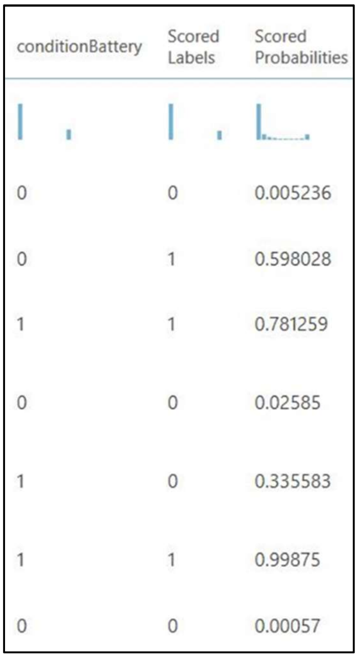
Setting option	**Selected Value**
Hidden layer specification	Fully-connected
Number of hidden nodes	100
Learning rate	0.1
Initial learning weights	0.1

Table 5-1: Selected settings of the Two-Class Neural Network

In all, 100 neurons are selected in the hidden layer. Despite there being only one hidden layer, a certain complexity should be allowed in order to uncover relationships between the variables. For the learning rate and the initial learning weights, the default setting is kept, since no values could be taken from the literature. This is 0.1 in both cases. The learning rate defines the size of the steps taken during the iterations to adjust the weightings of the connections between the neurons. A large learning rate can cause the model to converge faster, but it has the risk to overshoot a local minima. The initial learning weights just define the connection weights at the beginning of the learning process.

The next step is to train the model. For this, the previously defined ML method and the training data are used. Here the only variable in the data which has to be defined is the dependent one and it therefore has to be calculated as the output. All other variables are automatically used as input to calculate the output. The ANN then trains itself in this step. As explained in Section 2.3.4, this is done by adjusting the weights.

After the ANN has been trained with the training data, the values for the dependent variable of the 75,000 test data are calculated in the next step. This function block is named as 'Score Model'. The scored probabilities, the resulting label and the correct solution can be displayed as in Figure 5-2. Under the column headings, the distributions of the respective columns are shown. The calculated probability is compared with a threshold value. The default value is 0.5. The second entry represents an error. The scored label is a 1 (the battery has to be replaced), but the right label is a 0 (the battery has not to be replaced). But it turns out that the calculated probability of 0.598028 is not very far away from the threshold value.



conditionBattery	Scored Labels	Scored Probabilities
0	0	0.005236
0	1	0.598028
1	1	0.781259
0	0	0.02585
1	0	0.335583
1	1	0.99875
0	0	0.00057

Figure 5-2: Correct labels, scored labels and probabilities by the battery model

After the ANN has been trained, the model is adapted so that is possible to implement an interface to Python. The web service modules are provided by Microsoft Azure ML studio. Figure 5-3 shows the structure of the web service included in the model. In addition to the data used to train the artificial neural network, inputs from the executing program are also possible. In the next step, the necessary data is selected from the input data received. This step is analogous to the data selection in the previous model creation. Then the trained model for the respective component is used to determine the scored probability and the scored label to the input data. The generated information is then transmitted to the executive program via the web service. The deployment of the web services creates an application programming interface key (API key) and an URL address. The API key contains the necessary information to get access to the URL address of the referring web service model.

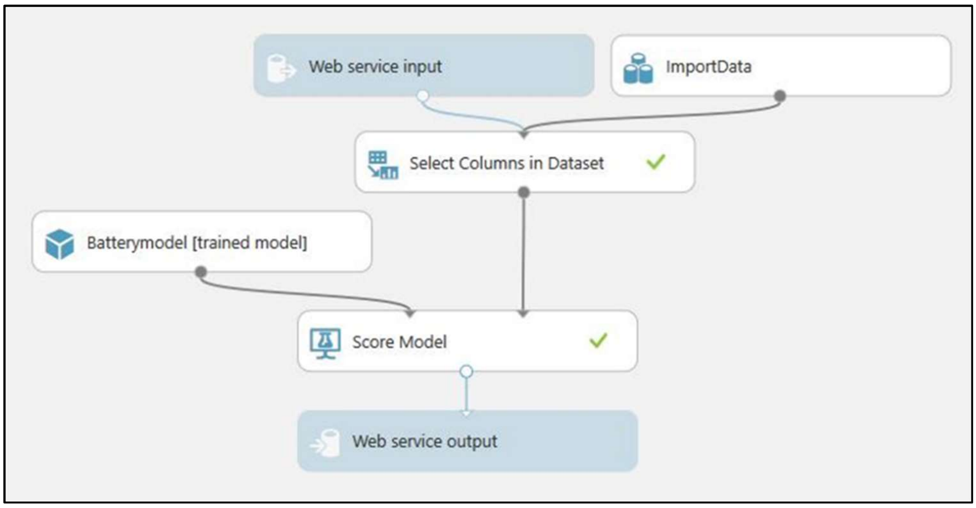


Figure 5-3: Implementation of web services

The described steps are also performed for the other nine components in order to train a model to determine the condition of the component.

5.2 Evaluation of the artificial neural networks

The last step is the evaluation of the trained model. Here the scored labels and probabilities from the previous step are used. The quality of the generated model is analysed and represented by various metrics.

Figure 5-5 shows the receiver operating characteristics (ROC) curve, which is used to evaluate the performance of classification models in ML (Bradley, 1997; Goncalves, Subtil, Oliveira, & De Zea Bermudez, Patricia de, 2014, p. 4). It indicates how many of the positive datasets were labelled as positive (true positive) and how many of the negative datasets were labelled as positive (false positive). The true positive rate (TPR) is given as a function of the false positive rate (FPR), which is due to the fact that the threshold can be shifted, which influences the rates. The ROC curve is used to calculate the area under the curve (AUC). This describes the ability of the model to separate the distribution of the two labels by setting a threshold.

Figure 5-4 shows the distributions of the probabilities of an optimal model. By defining a threshold between the two bells, each dataset could be assigned to the correct label. This means that the true positive rate would be 1 and the false positive rate 0. The ROC curve would thus run through the upper left corner. It would have the shape of a right angle and the AUC would be 1, which is the best possible value. If the threshold value would be moved to the left, into the red bell, the true positive rate would still be one, since all positive datasets would be labelled as positive. However, the false positive rate would no longer be 0, since negative datasets would also be labelled as positive.

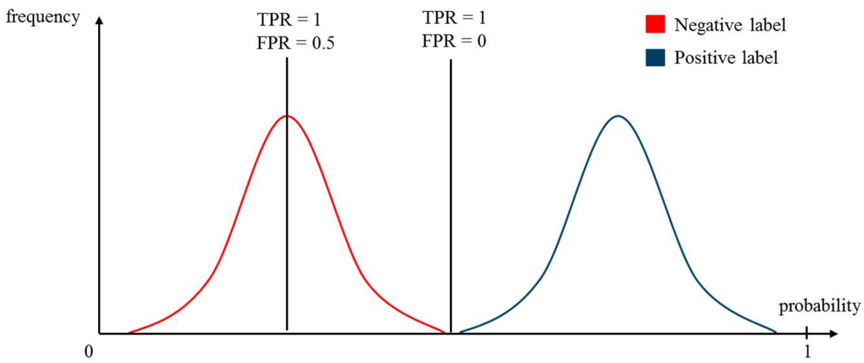


Figure 5-4: Distributions of calculated probabilities of an optimal model. Own illustration based on Goncalves et al., 2014, p. 4

If the datasets were to be labelled randomly, the chance of labelling the dataset correctly would be 50 % and the two bells would be on top of each other. The ROC curve would run as an angle bisector from the lower left corner to the upper right corner. The AUC would thus be 0.5, which is the worst value because it describes the coincidence. A value below 0.5 indicates a systematic error.

Figure 5-5 shows that the ROC curve of the generated model to determine the condition of the battery gets very close to the left upper corner, which shows that it has a high quality. Figure 5-6 shows that the model has an AUC of 0.93, which is a very good value.

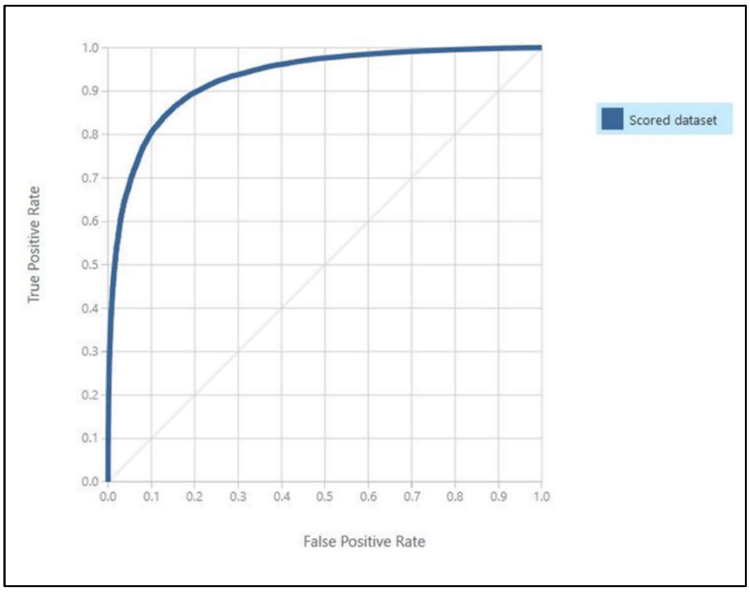


Figure 5-5: ROC Curve of created model for condition of battery

Figure 5-6 shows further metrics to measure the performance of the scored model with a threshold of 0.5. The precision is the proportion of true results over all as positive labelled results. 78.9 % of the batteries that the model designates as defective are actually defective or have to be replaced. The recall is the fraction of all positive datasets, which are labelled correctly. 71.3 % of defective batteries are detected as defective by the model. The accuracy measures the integrity of the classification model as the proportion of true results to total cases. The model assigns 88.8 % of the batteries to the right label. The F-score is the weighted average of precision and recall.

Figure 5-6: Metrics of the created model for condition of battery

A distinction of the errors is meaningful, since these have different consequences. For the system considered here, recall is the most important metric because the error that a defective battery considered as a functional one is the most serious error. This would lead to a laptop being priced at a higher value than it actually has and would lead directly to a loss of money. The error that a functional battery is considered to be defective is not so bad. This would lead to the laptop being assigned a lower value than it actually has. Only possible turnover is lost as a result but not the own money. Therefore, the precision and accuracy is subordinated to recall in this case. The recall value can be increased by shifting the threshold value. However, this will reduce the precision and the accuracy. Figure 5-7 shows the metrics of the model after changing the threshold from 0.5 (see Figure 5-6) to 0.25. The change in the threshold value leads to 84.9 % of defective batteries now being detected as defective. However, this also leads to only 65.2 % of the batteries designated as defective as actually being defective. The balance here has to be between laptops bought at too high price and lost sales.

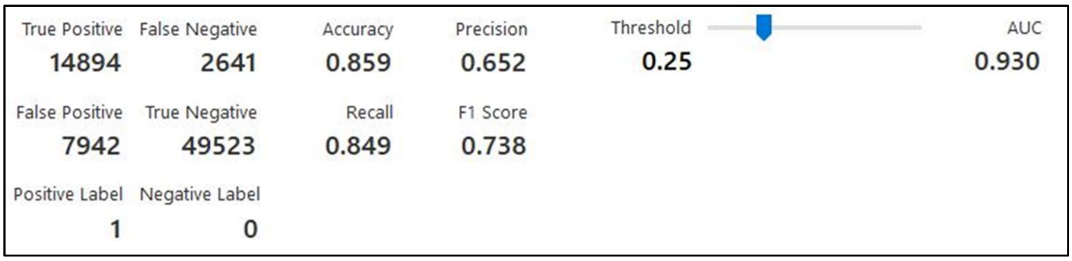


Figure 5-7: Metrics of the created model for condition of battery with adjusted threshold

Another possibility would be to introduce a third category, which is between defect and functional. So the components with a value between 0.3 and 0.7 could be checked manually. However, this results in higher costs and is difficult to implement due to the high number of laptop components. The probability that one of the components has a value in this range is very high and so again all laptops would have to be sent in and checked manually.

Table 5-2 lists the important metric of all trained models.

Table 5-2: Metrics of trained models

Component	AUC	Threshold	Accuracy	Recall	Precision
Battery	0.930	0.25	0.859	0.849	0.652
CPU	0.953	0.25	0.927	0.851	0.695
GPU	0.941	0.25	0.936	0.719	0.678
Display	0.973	0.45	0.929	0.851	0.854
Keyboard	0.983	0.5	0.956	0.870	0.885
Touchpad	0.942	0.25	0.925	0.731	0.673
Motherboard	0.932	0.25	0.902	0.713	0.669
Housing	0.964	0.25	0.921	0.848	0.716
Hard drive	0.887	0.25	0.887	0.610	0.508
Cooling system	0.940	0.25	0.883	0.817	0.717

As previously explained, the AUC describes the model’s ability to separate the two labels. Accuracy, recall and precision, on the other hand, depend on the defined threshold value. This threshold is chosen so that it does not get smaller than 0.25, so that not all components are simply seen as defective. Furthermore, it was set to the highest possible value, so that the recall is still larger than 0.85. This prevents many laptops which could be bought economically from being rejected. It shows that the models of all components have very good AUC values. The hard drive is the only outlier that stands out. It turns out that the components, from which a first impression can be obtained without disassembly can especially be determined very well. This is shown by the high recall values although the threshold value for the display, keyboard and housing is higher. Only the touchpad has different values. Also the condition of the battery and the CPU can be determined very well, since both have a recall value of about 0.85. As the AUC has already shown, the hard disk model has the worst metrics. The accuracy is even better than the one of the battery and the cooling system, but the recall is much worse. This means that it detects the functional hard drives very well, but only 61 % of the defective hard drives. Therefore, this model should be trained again with different configurations or new independent variables should be introduced which describe the condition of the hard drive. Since the doubling of the learning iterations and the increase of the neurons in the hidden layer up to 200 have also lead to an AUC of 0.866 and a recall value of 0.595 at a threshold value of 0.25, new independent variables should be used.

Figure 5-8 points out the problem of the hard drive model. It shows the histograms of the scored probabilities of the models of the battery and the hard drive. It can be clearly seen that the battery model has, besides having most values in the range 0.0 to 0.2, also a frequent occurrence at the end of the scale between 0.9 and 1.0. This concentration is not found with the model of the hard drive. This means that the model cannot identify datasets accurately as obvious defective laptops. Therefore, the threshold must be lowered so far to get a recall of just 0.61. This means that 61 % of the defective laptops are on the right side of the threshold of 0.25. However, 88.7 % of all laptops are on the correct side of the threshold.

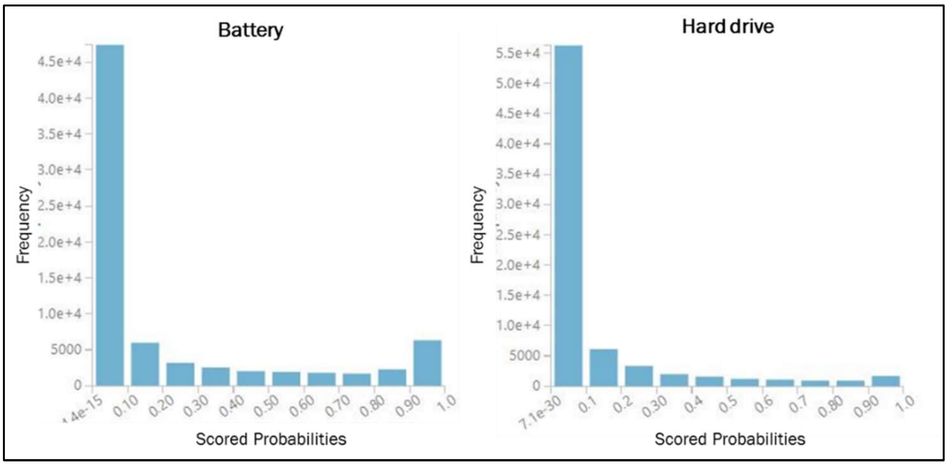


Figure 5-8: Histograms of the scored probabilities

5.3 Implementation of other machine learning classification methods

After a detailed description of the implementation procedure and the results obtained, two further classification methods are used. A support vector machine and decision tree were applied to determine the condition of the laptop components. The implementation in Microsoft Azure ML Studio was analogous to the ANNs. Figure 5-9 illustrates the ROC-curve and the important metrics of the two ML methods, without adjusting the threshold.

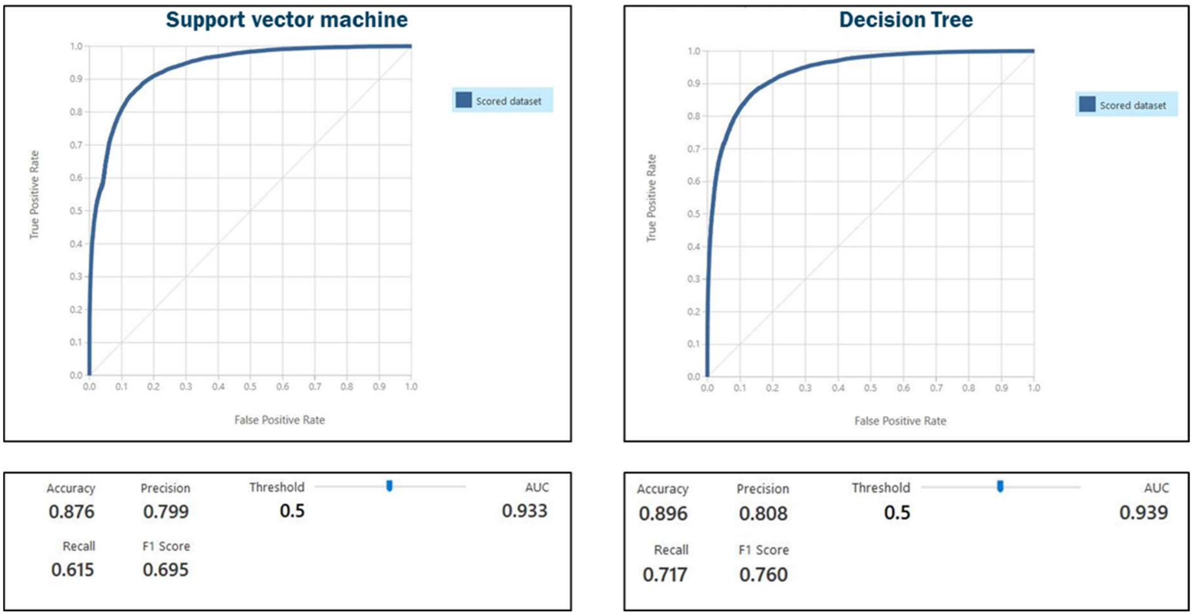


Figure 5-9: ROC-curve and metrics of other classification methods

The curve is similar for both methods and the previously presented ANN, which is also reflected in the AUC value. However, the recall value of the support vector machine is much lower. This suggests that this method is not suitable for the processing of the existing data. When processing real data, the different methods should be tested and the key figures compared. Since these were not generated with a mathematical model, they have much more complex relationships. It is therefore necessary to test again which method is most suitable. In addition, requirements must be defined with regard to the available data volume and available computing power. If only few data are available for training the model ANNs could lead to worse results than other methods as they are quite comprehensive and powerful.

5.4 Implementation of economic and ecological evaluation

This section deals with the implementation of the economic and ecological, which have been developed in Section 3.7. The executing program is coded in Python.

5.4.1 Structure of program

Figure 5-10 and Figure 5-11 illustrate the class diagrams of the developed decision-making process in Unified Modeling Language (UML) format. Class diagrams describe the structure and architecture of a software system and deal with the first notation for object-oriented-modelling (Rumpe, 2016).

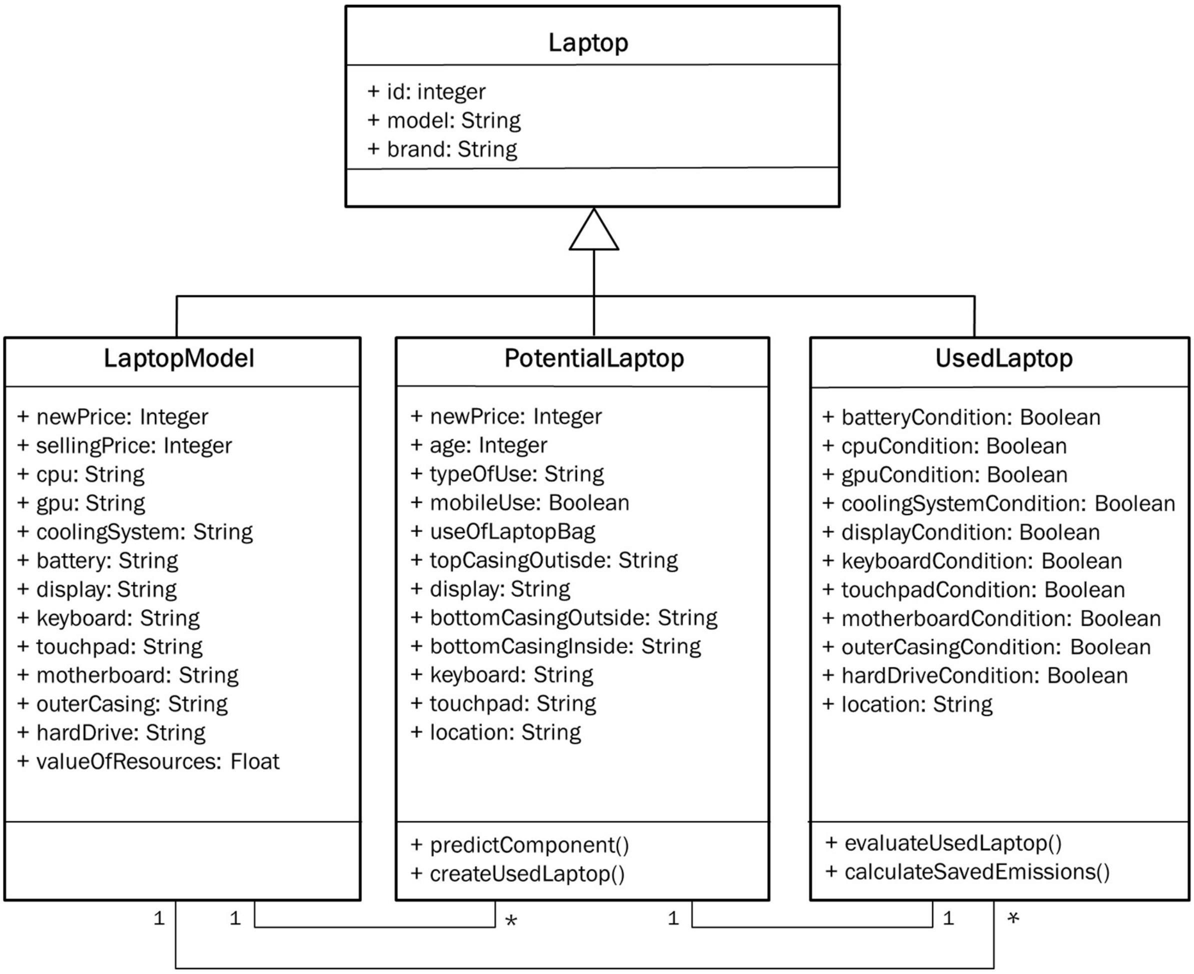


Figure 5-10: UML class diagram of the laptops for the evaluation system

In the UML diagram a class is shown as a box. A class consists of three elements: name, attributes and methods. The class name identifies the class and is stated in the first compartment of the class representation. The attributes and methods describe the state and behaviour of the instances of a class. These are generated objects of the class. An attribute is described by its name and type. They form the middle compartment in the illustration. The type defines which values the attribute can assume. The last compartment shows the methods of a class. The functionalities of a class are stored in methods, which serve the execution of tasks and data processing. Furthermore, relationships between classes can be established. One of them is inheritance, which is illustrated as a connection line with a triangle at one end. The triangle indicates the superclass, the other class forms the subclass. The subclass inherits all attributes and methods of the superclass. Further connection lines can represent associations between objects of classes. The cardinalities at the end of the lines describes whether an association between objects is optional or mandatory, or allows many bindings. (Rumpe, 2016, p. 17)

Figure 5-10 illustrates the UML class diagram of the part of the software regarding the different laptop classes. Therefore, a super-class "Laptop" is created, since all laptop classes have the attributes id, model and brand. The system distinguishes between three different laptop types. The first subclass class comprises the laptop models under the class name "LaptopModel". An instance of this class is a particular laptop model and therefore contains attributes that provide information about the price, used components, residual value of resources and associated emissions. The information, which is required to create the instances of this class, is stored in the system and is available from the beginning, However, the program can easily be extended by models.

The second laptop subclass describes the laptops which will be evaluated, under the class name "PotentialLaptop". An instance of this class is only created by the input of information from a user of the system. Here the owner of an EoU laptop enters the required information for the laptop. This is the information which is used later to determine the condition of the components. Figure 5-10 shows the cardinalities of the association between "LaptopModel" and "PotentialLaptop". A potential laptop can only be assigned to one certain laptop model, but a laptop model can be assigned to many different laptops under review. An association between two different sets of objects is achieved by means of keys, which are certain attributes of the objects. A primary key is an attribute whose values occur only once in a set of objects and are therefore uniquely identifiable. A foreign key is an attribute in which values occur more than once, but can assume the same values as the primary key of another object set. This means that each object in the set of the foreign key can be assigned one object from the set of the primary key (Unterstein & Matthiessen, 2012, p. 29). There the primary key is the attribute *model* in the class "LaptopModel" and the foreign key is the attribute *model* in the class "PotentialLaptop". In addition, the class contains the function *createUsedLaptop()*, which triggers the communication with Microsoft Azure ML Studio. It calls the methods *predictComponent(),* which transmit the data of the object to the respective ANN, which determines the condition of the component and returns it as a response. For each component a separate *predictComponent()* method is defined. The structure und functionality of the method will be discussed in more detail later. The createUsedLaptop() method creates an object of the class "UsedLaptop" with the received information.

The "UsedLaptop" contains the determined condition of all components. Each object of the class "UsedLaptop" can only be assigned to one object of the class "PotentialLaptop". This also applies the other way round. For this reason, both keys used must be primary keys and therefore the ID is used for the unique assignment. The association to the class "LaptopModel" is generated similarly to the association from "LaptopModel" to "PotentialLaptop". The attribute *model* is used as key. In addition, the class contains the methods *evaluateUsedLaptop()* and *calculateEmissions().* The first method performs the economic evaluation of the laptop and responds with a price for which it can be bought from the owner. The method *calculateEmissions()* delivers the amount of CO_2 emissions are saved by reusing or remanufacturing the EoU laptop.

Besides the classes of laptops there is also the class "Repair". In addition, further methods and variables, which are not assigned to a specific class, are defined as global in the executive program.

Figure 5-11: UML class diagram of repair, location and global variables

The class “Repair” contains all necessary information regarding repairs. A separate object is created for each component of each laptop model. The variables *labourCosts* and *repairCosts* are determined using the defined class methods. There are two ways to determine the value of the attribute *repairCost* of a repair. Either it is determined as the sum of *sparePartPrice* and *labourCost* or a fixed value is defined when the repair is created. Global methods are used to access variables of a class from outside. The global variables are easily accessible and can be changed centrally. The class for creating the GUI is not discussed here, since it only serves to increase user-friendliness and the structure is mainly given.

5.4.2 Programming

The entire source code can be found in Appendix B.

At the beginning, the necessary libraries *urllib.request* and *json* are imported. These are required for the communication with Microsoft Azure ML Studio. Next, the global variables are then defined. In the next section, the classes are defined, as describes previously. Table 5-3 shows a part of the source code of class ‘LaptopModel’

Table 5-3: Source code of class “LaptopModel”

```
class LaptopModel(object):
  def __init__(self, id, model, brand, price, sellingPrice, cpu, gpu
               coolingSystem, battery, display, keyboard,
               touchpad, motherboard, outerCasing, harddrive):
      self.ID = id
      self.Model = model
      self.Brand = brand
      self.NewPrice = price
      self.SellingPrice = sellingPrice
...
```

The *__init__(...)* method describes the constructor with which instances of the class are created. The values, which are in the brackets when creating an instance, are assigned to the attributes of the object in the order as defined in the constructor. The statement *self.Attribute=attribute* in the constructor is repeated for all attributes of the class. The classes “PotentialLaptop”, “UsedLaptop”, and “Repair” are defined similarly as shown in Table 5-4.

Table 5-4: Source code of classes “PotentialLaptop”, “UsedLaptop” and “Repair”

```
class PotentialLaptop(object):
   def __init__(self, model, age, typeOfUse, mobileUse, useOfCase,
                newPrice,brand, topCasingOutside, display,
                bottomCasingOutside,bottomCasingInside,
                keyboard, touchpad, location):
...

class UsedLaptop(object):
   def __init__(self, id, model, brand, cpuCond, gpuCond,
                coolingSystemCond,batteryCond, displayCond,
                keyboardCond, touchpadCond, motherboardCond,
                outerCasingCond, harddriveCond):
...

class Repair(object):
   def __init__(self, rid, model, component, sparepartprice,
                labourhours, repairCost):
...
```

After the classes are defined, the objects of the classes “LaptopModel” and “Repair” are created. The information about the laptop models has been taken from online available data sheets or online offers and partly filled with dummies, or no clear designation of the component was found. The information on the repairs was also researched online. As already mentioned, the repair costs can either be calculated on the basis of the spare part price and the required labour time, or were taken over by a repair provider at a fixed price. Table 5-5 illustrates the source code of creating one laptop model instance and two repair instances.

Table 5-5: Source code of creating laptop model and repair objects

```
lifebookA357 = LaptopModel(0, "Lifebook A357", "Fujitsu", 479, 299,
                "8 GB DDR4 2133 MHz SODIMM", "Intel HD Graphics 620",

                "CoolingSystem LifebookA357", "Battery LifebookA357",
                "Display LifebookA357","Keyboard LifebookA357",
                "Mousepad Lifebook A357", "Motherboard LifebookA357",
                "Casing LifebookA357", "256 GB SSD SATA III")
...

repairLifebookA357CPU = Repair(0, "LifebookA357", "CPU", 56, 0.7, 0)
repairLifebookA357GPU = Repair(1, "Lifebook A357", "GPU", 0, 0, 399)
...
```

The instance *lifebookA357* is created by calling the constructor of the class “LaptopModel”. The constructor is filled with the values for the attributes of the instance. An object is created for all laptop models, which should be included in the system. A new model can be added here later using the constructor. However, it should be possible to add a model using external input outside the source code. Similarly, an instance of the class ‘Repair’ is created for each laptop model and component.

The program is started by the call of the application user. This executes the method *createPotentialLaptop()*. Table 5-6 illustrates the source code structure of the method, which is used to create an object of the class "PotentialLaptop".

Table 5-6: Source code of method createPotentialLaptop()

```
def createPotentialLaptop():

    potentialLaptop = PotentialLaptop(inputModel, inputAge,
                                      inputTypeOfUse,inputMobileUse,
                                      inputUseOfCase, inputNewPrice,
                                      inputBrand,inputCoverOutside,
                                      inputDisplay,inputBaseOutside,
                                      inputBaseInside,inputKeyboard,
                                      inputTouchpad, inputLocation)
```

This method uses the information of the laptop owner entered via the GUI. After completion of the input it is used in the call of the constructor of the class "PotentialLaptop". This creates an instance, which represents the EoU laptop under review.

After the object of the EoU laptop has been created, the condition of the components of the laptop is determined on the basis of the information received and the already trained ANN. Method *createUsedLaptop()* is executed therefore. Table 5-7 shows the relevant part of the source of this method.

Table 5-7: Source code of method createUsedLaptop()

```
def createUsedLaptop(potLaptop)

    usedLaptop = UsedLaptop(1, potLaptop.Model, potLaptop.Brand,
                         predictCPU(potLaptop),
                         predictGPU(potLaptop),
...
```

When the method is called, the previously generated object *potentialLaptop* is passed. The passed object is referred to as *potLaptop* in the course of the method. In order to create the instance of the class 'UsedLaptop' the corresponding constructor is called. The constructor contains a defined id, as well as the model, brand and location of the passed object *potLaptop.* The remaining constructor attributes, which describe the conditions of the components, are determined by the method *predictComponent().* When calling the method, the object of the potential laptop is transmitted, since the information is necessary for the determination of the conditions. This method is called for each component, since different ANNs are used.

The communication between the executing program and Microsoft Azure ML studio uses Representational State Transfer – Application Programming Interface (REST-API), which is applied in the method *predictComponent()*. This makes it possible to exchange information when it is on different systems, since it allows machine-machine communication. For each component, this method is modified because it contains different information. Table 5-8 illustrates the relevant parts of the *predictComponent()* method at the example of the component battery.

Table 5-8: Source code of method predictComponent()

```
1    def predictCPU(potLaptop):
2         data = {
3           "Inputs": {
4              "input1":
5                 [ {
6                      'model': potLaptop.Model,
7                      'age': potLaptop.Age,
8                      'typeOfUse': potLaptop.TypeOfUse,
9                      'mobileUse': potLaptop.MobileUse,
10                     'useOfCover': potLaptop.UseOfCase,
11                     'price': potLaptop.NewPrice,
12                     'brand': potLaptop.Brand,
13                     'coverOutside': potLaptop.TopCasingOutside,
14                     'display': potLaptop.Display,
15                     'baseOutside': potLaptop.BottomCasingOutside,
16                     'baseInside': potLaptop.BottomCasingInside,
17                     'keyboard': potLaptop.Keyboard,
18                     'touchpad': potLaptop.Touchpad,
19
20   ...
21
22        body = str.encode(json.dumps(data))
23        url = 'https://ussouthcentral.services.azureml.net/ ...
24        api_key= 'T/xkbzVgl1TiClr/GA7o+KXaJXAay8JhvuPwE255LiDyUwt ...
25        headers = {'Content-Type': 'application/json', 'Authorization':
26                     'Bearer  ' + api_key)}
27        req = urllib.request.Request(url, body, headers)
28
29        try:
30             response = urllib.request.urlopen(req)
31             result = response.read()
32             x = json.loads(result);
33             r = x["Results"]['output1'][0]['Scored Labels']
34             if r == "1":
35                 return (1)
36             else:
37                 return (0)
38
39   ...
```

In the first part of the method, the data to be transmitted is defined. Lines 6 to 18 show the assignment of the attributes of the passed object *potLaptop* to column names of the data used in Microsoft Azure ML Studio. The information for the communication between the executive system and the ANN is then defined in the lines 22 to 27. The method variable *url* is the address of the required ANN. It is therefore different for all components. The *api_key* is also different for each component and forms the password with which the executing system gets access to the ANN. The third part (line 29 to 37) describes the processing of the received response. The response is first stored and then read. Then the delivered value of the attribute ‘Scored Labels’ is cached. This contains the information of the calculated condition of the component. Since it is a variable of the type String, a comparative query is carried out in order that the method *predictComponent()* is able to return a binary answer. This information is then used in the constructor to generate the object.

Finally, the previously generated object, which contains the information about the conditions of the components, is used to perform the economic evaluation. This is defined in the method *evaluateUsedLaptop(),* which is shown in Table 5-9.

Table 5-9: Source code of method evaluateUsedLaptop()

```
def evaluateUsedLaptop(usedlaptop):
  sellingPriceLaptop = getSellingPrice(usedlaptop)
  savedEmissions = calculateSavedEmissions(usedlaptop)
  valueResources = getValueResources
  repairCosts = calculateRepairCosts(usedlaptop)
  refurbishCosts = hourlyRateRefurbishing * labourHoursRefurbishing
  profit = sellingPriceLaptop * profitMargin
  riskAddOn = sellingPriceLaptop * riskProportion
  capitalCommitmentCosts = (sellingPriceLaptop - profit - riskAddOn
                            - repairCosts - refurbishCosts)
                            *(interestRate * averageStorageTime)
  sumcosts = round(profit + riskAddOn + capitalCommitmentCosts
                 + repairCosts + refurbishCosts, 0)
  possiblePurchasePrice = sellingPriceLaptop - sumcosts
  recyclingValue = getValueOfResources(usedlaptop)
  if ((possiblePurchasePrice) <= 0):
       print(textforPurchasing)
  else:
       print(textForRecycling)
```

First, the selling price of this laptop is determined via the foreign key *model.* Subsequently the saved emissions are calculated by the method *calculateSavedEmissions().* It takes the value of the saved emissions for a complete laptop and subtracts from it the emissions caused by the required spare parts. For this it runs through the conditions of the components of the passed object *usedlaptop.* Afterwards, the repair costs incurred are determined by a separate method *calculateRepairCosts().* This method calculates the repair costs for each component that must be replaced by running through the objects of repairs and accessing their information. The remaining costs drivers are then calculated and totalled. The possible purchase price of the EoU laptop is a function of the total costs and the selling price. If this greater than zero, the user receives a message that the laptop will be purchased and the corresponding price. If the possible purchase price is less than zero, a message is issued that the laptop can only be purchased for recycling and the possible price for which it can be purchased. This value is based on the remaining value of the components contained in the laptop.

In order to make it easy to enter the required information, a GUI is created. Figure 5-12 represents a first draft of the GUI.

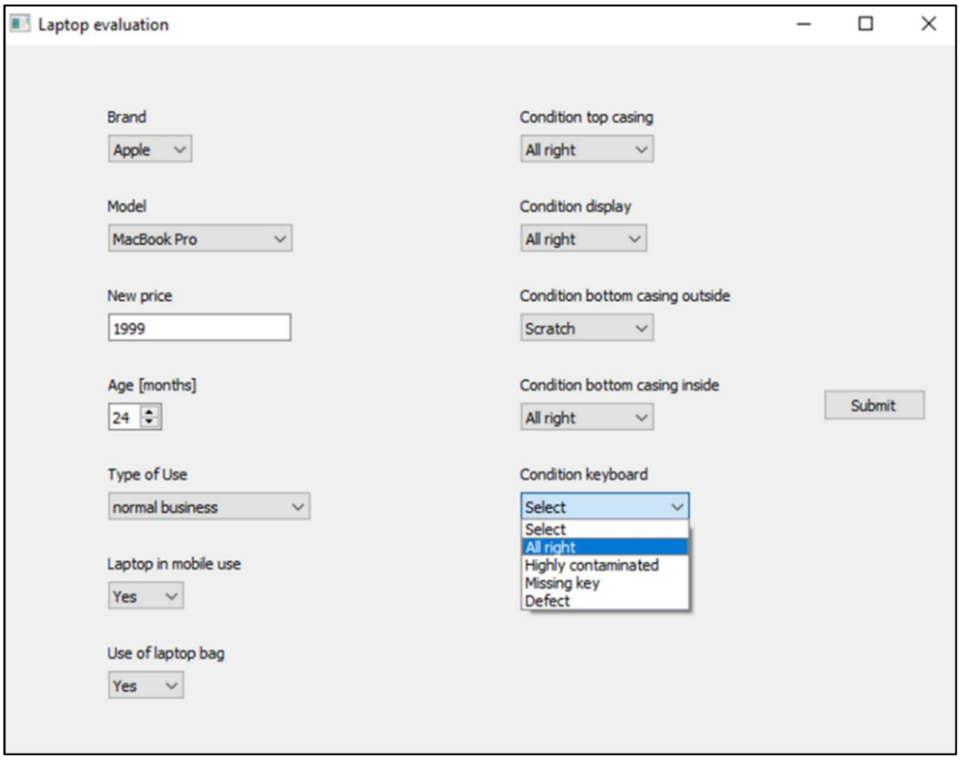


Figure 5-12: Graphical user interface

The categorical input information is captured via checkboxes, which are located under the corresponding label. By clicking on the arrow symbol, all possible input options are suggested and the user can select the respective one (see Figure 5-12 checkbox 'Condition keyboard'). This simplifies input for the user and prevents invalid input. The age of the EoU laptop is gathered via a spinbox. Arrow keys can be used to increase or decrease the integer value. In addition, direct input via the keyboard is possible. Only integer values are allowed. The price is recorded via a text edit window. This is done via keyboard input as well. An integer validator is applied in order to prevent invalid entries, such as letters.

Once all information has been entered, it can be confirmed using the 'Submit' button, which triggers the decision-process.

Figure 5-13 shows the possible output of the executive programme, which is sent to the user as a message.

```
We would like to buy your laptop for 265 €.
You will save 176 kgCo2eq by returning your laptop.


Unfortunately, your laptop cannot be reused or remanufactured.
However, we would recycle your laptop for you and pay you 40 € for the resources it contains.
```

Figure 5-13: Output of the executive programme

The upper output shows the case that the laptop can be recovered economically. Therefore, the difference between the expected selling price and costs incurred is offered to the owner of the laptop. In addition, customers are informed how many emissions they are likely to avoid by returning their laptop.

The second output shows the case that the laptop cannot be recovered economically. This is communicated to the customer. However, the customer is offered to recycle the laptop and pay a price for the resources contained in the laptop.

5.5 Chapter summary

This chapter serves the implementation of the prototype of the decision-making process based on the theoretical preparatory work of the previous chapters. First, the ANNs are implemented in Microsoft Azure ML Studio to determine the conditions of the laptop components. Then the performance of the ANNs is evaluated using different metrics. Small changes are made to the settings of the models to optimise the results. The second part of the developed decision-making process, which also serves as executive system, is implemented in Python. For this purpose, the structure of the programme is first defined and then implemented.

Chapter 6
Verification and validation

After the developed decision-making process has been implemented, it goes through a validation and verification process. The methodology used will be described first. Verification of a system component means checking the consistency between the component and its specification, which answers the question whether the component was developed correctly (Alpar et al., 2014, p. 317). Validation refers to the suitability of a system component in relation to its intended use and answers the question of whether the correct component has been developed (Alpar et al., 2014, p. 317).

6.1 Methodological approach

At the beginning of this work an iterative variant of the waterfall model was introduced (see Section 1.4). Validation and verification, however, follows the V-model described by Boehm (1979). It puts more emphasis on verification and validation compared to the waterfall model. Therefore, it divides the single step "Testing and evaluation" into several sub steps, which is the reason why it is applied. Figure 6-1 represents the V-model.

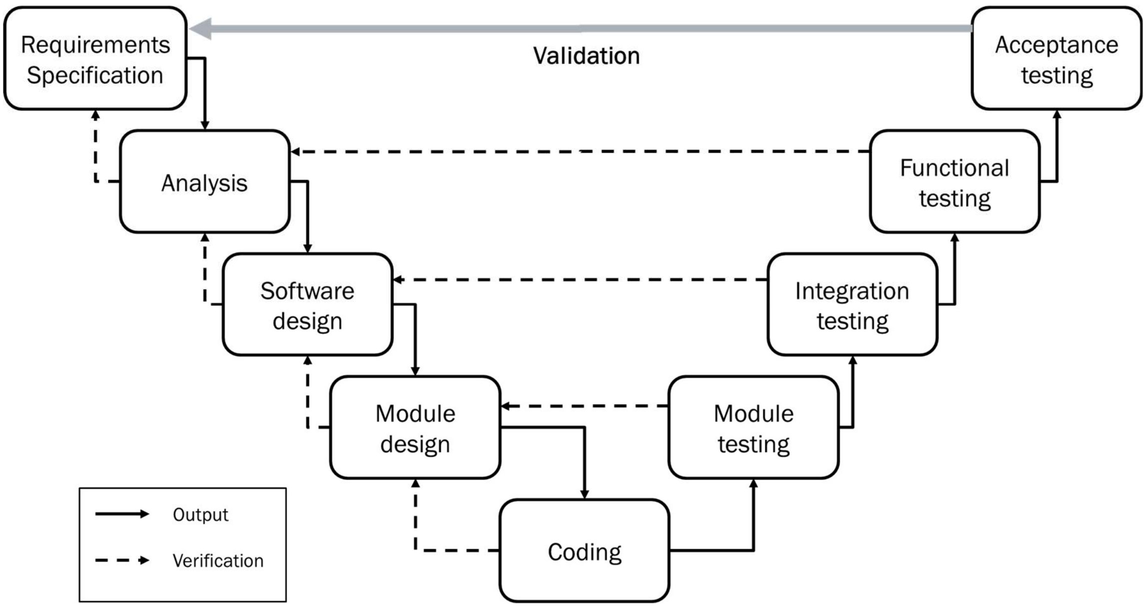


Figure 6-1: Development life cycle of the V-Model. Own illustration based on Kneuper (2018, p. 84)

The left-hand, descending branch represents the steps of the waterfall model. In this work, the individual steps were run through, whereby feedback to previous steps was possible if errors or potential for improvements were noticed during verification. Verification was performed step-by-step for newly developed parts. Subsequently, the developments and designs were implemented in the step “Coding” in order to test them. The individual modules, ANNs for condition determination and economic and ecological evaluation, were first programmed and tested. They were then integrated into one program. The verification of the coding steps was thus always carried out in parallel with the development and implementation. Finally, it is validated whether the entire system meets the individual requirements.

6.2 Verification

The verification of the individual steps has always been carried out by the completion of a component or module. The verification arrows in the left descending branch in Figure 6-1, describe adjustments or confirmations of previous steps. This is made possible because with the processing of the following step the understanding of the topic increases and by the higher degree of detail the degree of abstraction decreases and thus problems are recognised more easily. These are adaptions of the input variable types, since the selected type cannot be processed by the selected ML method. Newly gained insights were also incorporated into previous steps.

With the actual implementation of the theoretically developed program, the individual development steps can be checked again from another perspective. This describes the connection between the left and the right-hand branch of the V-model. The main purpose of this is to check whether the implementation is working as specified and defined. Table 6-1 lists and describes the conducted verification steps.

Table 6-1: Verification of development steps

No.	Development step	Verification
1.	Module design	Condition determination: Analysis of the metrics provided by Microsoft Azure ML Studio. Plausibility check of individual data sets. Economic and ecological evaluation: Check whether a reasonable response to manually generated extreme cases is provided.
2.	Software design	Test of communication of the two modules in the executing program.
3.	Analysis	Complete test runs from GUI input of the information and output of the calculated response, which was then checked for plausibility.

After coding, the design of the two modules was verified first. The metrics provided by Microsoft Azure ML Studio were used to verify the determination of the conditions of the laptop components. These were discussed in Section 5.2. In addition, individual datasets were randomly checked for plausibility in order to rule out systematic errors. After that the economic and ecological evaluation module was checked to verify whether the defined classes create objects correctly and whether functions terminate. In addition, functions were processed and the variables, which are to be changed were permanently printed in order to be able to reproduce the individual program steps. Furthermore, laptops were created manually and they went through the whole evaluation process. Extreme cases were used to check whether the response received changed in the right direction.

In the next step, the two modules were integrated into one program. The established communication interface was tested with test cases for each ANN. Attention was paid to whether the correct information was transferred to the network to be reviewed and whether a valid response was provided. Finally, the functionality of the entire system was examined. Complete test runs of the system were carried out for this purpose. The GUI was used to enter examples and the received response was checked for correctness. This ensures that all components of the system are running correctly.

6.3 Validation

Following the verification of the implemented decision-making process, it is validated in two stages: the fulfilment of the defined requirements and a scenario analysis. These are discussed in the subsections that follow.

6.3.1 Validation of user requirements

In order to validate the results of the implementation of the defined requirements in Section 3.2, two short anonymous surveys were conducted. In the first survey, eleven potential users of the developed decision-making process were asked for their assessments. Users here are laptop owners who could use the process as customers. Because of the low number of participants, the survey is not a completely representative survey. But it shows a first impression. The survey consists of six questions with single selection. A summary of the project was given to the participants in the form of an information mail. They also received two short videos showing demonstration runs of the prototype. The information letter is represented in the Appendix C.

The results of the first two questions of the survey are shown in Figure 6-2.

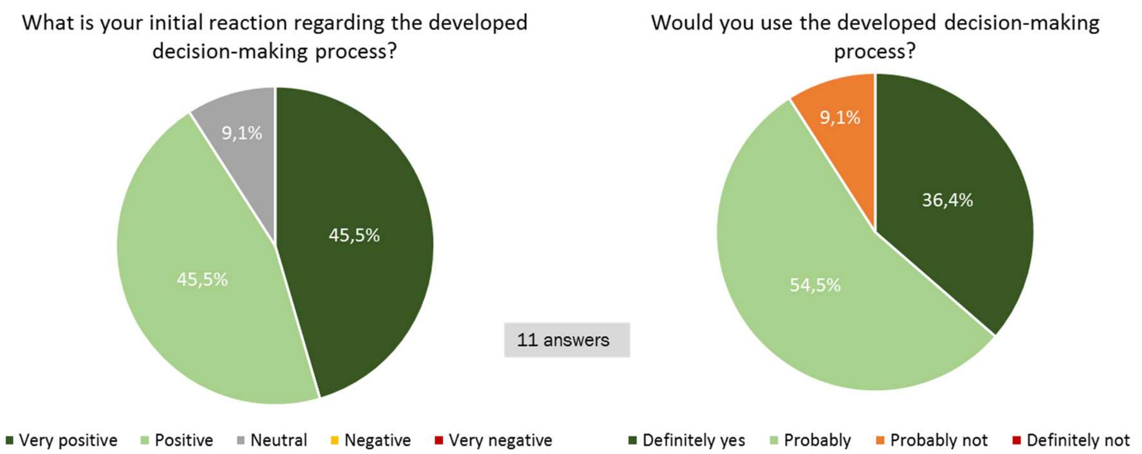


Figure 6-2: Opinion of potential users on the developed process

At the beginning of the survey, a first impression of the entire decision-making process and the presented prototype shall be recorded. The pie chart shows, that the participants react predominantly positively to the developed process.

The next question goes one step further and asks if the process would also be used. The result remains still positive, although not quite as positive as the first impression. However, the majority would probably use the decision-making process.

The third and the fourth questions serve to determine the perceived attractiveness of the developed decision-making process. Figure 6-3 shows the results of these questions.

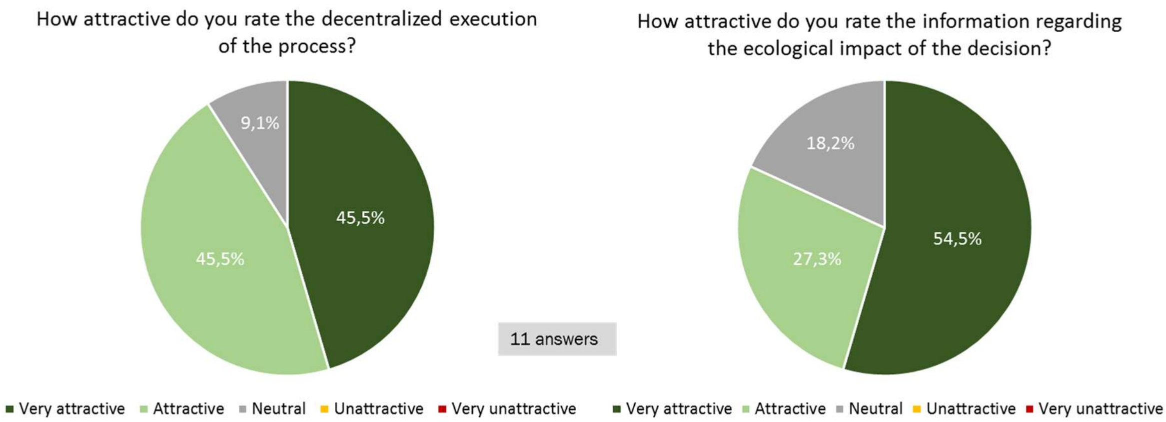


Figure 6-3: Attractiveness of the process for potential users

It shows that the decentralised execution of the process by the user is perceived as attractive. The decision-making process is conducted locally by the owner of the EoU laptop, without the need to send the laptop to a collection institution. This reduces the effort for the user and the feedback is available faster than if the laptop has to be sent in.

The results of the fourth question show that the information on the ecological impact is also positively received. This is also reflected in the general public awareness of sustainability issues. However, it can be recognised that there are also users who are indifferent to this topic and who are presumably primarily motivated by the financial benefits.

The last two questions of the survey refer to the user-friendliness of the developed process. Figure 6-4 shows the results.

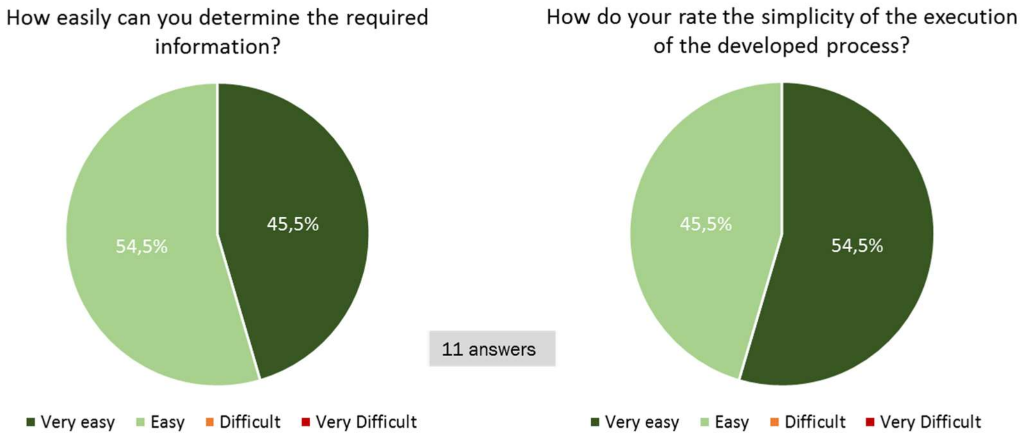


Figure 6-4: User-friendliness of the process

The respondents can determine the information required for the process easily or very easily. This shows that the requirement to select information that can be obtained without effort is met.

The general execution of the process is perceived even more easily. The developed GUI allows a fast, intuitive and trouble-free input of the information. Access to the system could easily be achieved via a smartphone application or homepage.

In summary, it can be seen that the developed decision-making process is well received by the participants, which represent potential users.

6.3.1 Validation by expert survey

In addition to the survey for potential users of the developed decision-making process, a survey with experts from the fields of IT, CE and business solutions was conducted. Table 6-2 lists the expertise of the individual experts. The possibility of implementation and benefit of the developed process should be evaluated.

Table 6-2: Expertise of the experts

No.	Expertise
1.	Expert for IT solutions
2.	Expert for sustainable recycling management
3.	Expert for function development and model-based software development
4.	Expert for business solutions
5.	Expert for data analytics

At the beginning of the survey, the assessment of the economic and ecological relevance of the topic is asked, which is illustrated in the left pie chart of Figure 6-5.

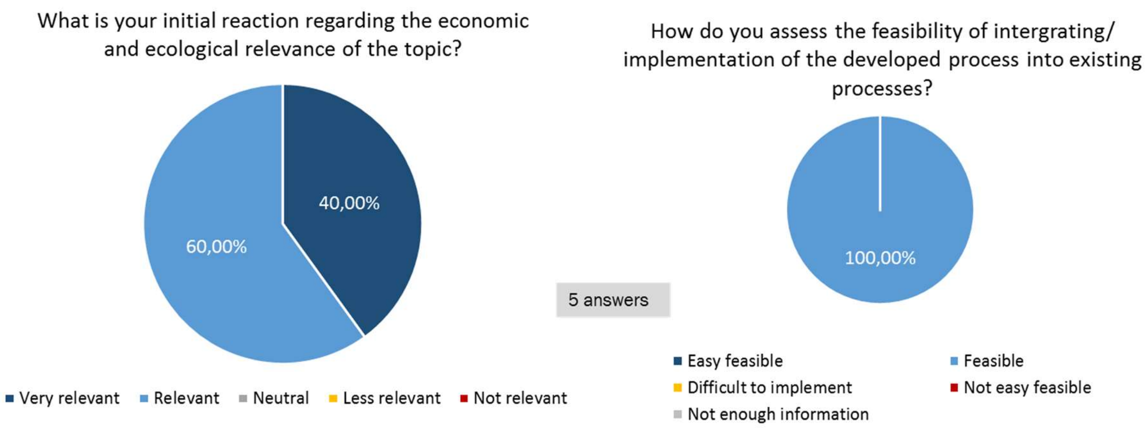


Figure 6-5: Experts opinions on the relevance of the topic and feasibility of the process

The experts consistently assess the topic to be at least relevant. This shows that this topic is currently being focused and should be further researched.

Next, the feasibility of implementing the developed process is assessed. The experts are in agreement and all consider the implementation to be feasible. No one thinks the implementation is easy. This shows that specific know-how is required with regard to CE and ML.

The next two questions deal with the selection of the ML method and the information used to determine the conditions of the laptop components. The results of these questions are illustrated in Figure 6-6.

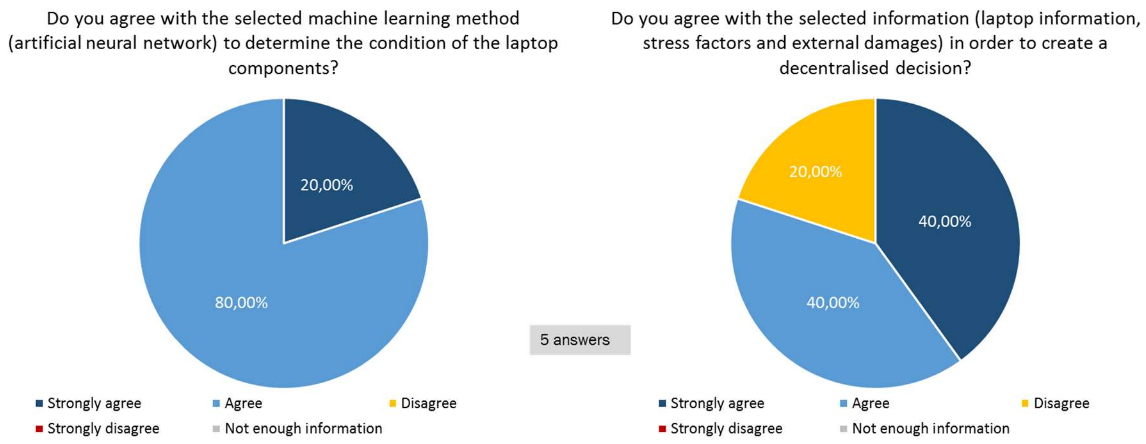


Figure 6-6: Assessment of the selection of ML method and information used by experts

The experts consistently agree to the selection of ANNs as used method to process the recorded data. One expert does not agree to the selection of the information used to determine the component conditions. Therefore, further information should be considered that could provide an indication of the condition of the laptop. The use of information that require an upgrade or specific development of the laptop was excluded in this work, as the process should be also applicable to products not designed for the CE. However, further information should also be taken into account in the future.

Figure 6-7 shows the results of the questions regarding the expected economic and ecological benefits of the developed decision-making process. The experts assess both the economic and the ecological benefits as positive to very positive. In particular, economic benefits are a driving factor for implementation, as defined in the requirements for the process. The ecological aspect can increase the customer's motivation or play an important role through governmental regulations.

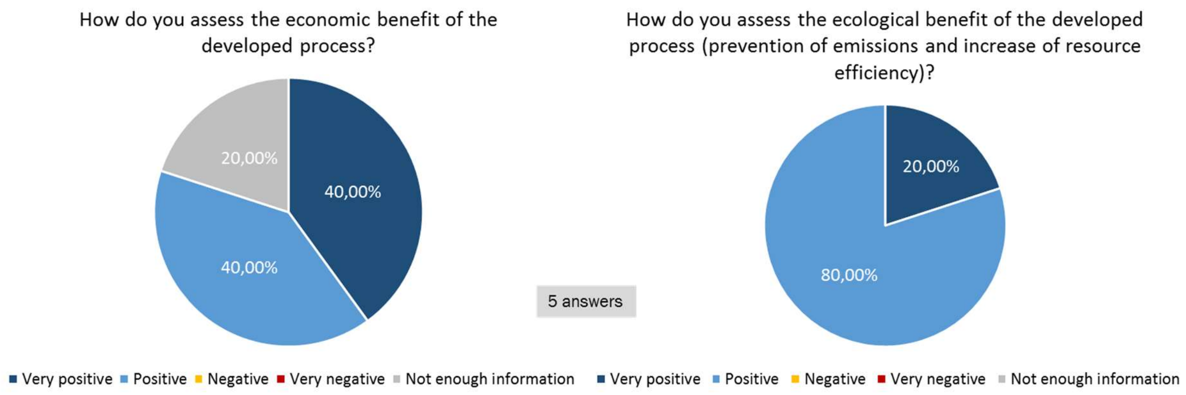


Figure 6-7: Assessment of the economic and ecological benefits by experts

Next the ability of the process to deal with other products is evaluated. A distinction is made between other electronic and non-electronic products. The results of these two questions are illustrated in Figure 6-8.

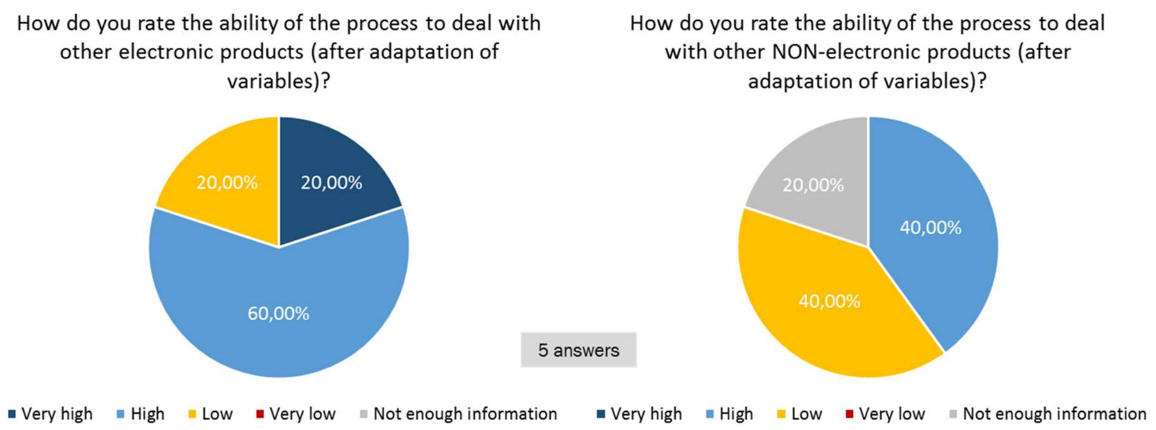


Figure 6-8: Assessment of the adaptability of the process to other products by experts

The experts consider the ability of the process to deal with other electronic products to be high. In particular smartphones and tablets are used very much like laptops. They also react similarly to the stress factors. Small adjustments of the input and output variables should already make it possible to deal with these products. This includes, for example, the absence of fans in smartphones.

The experts are much more critical of the ability of the developed process to deal with non-electronic products. This is presumably due to the fact that significantly larger changes have to be made. Furthermore, it is more difficult to record information by sensors. In addition, electronic products are among the more expensive products. In case of non-electronic products, the high investments may not pay off.

Finally, the experts evaluate the attractiveness of the developed decision-making process for the customers. Figure 6-9 shows that the assessment of the experts' assessment is similar to that of the customers themselves, which was previously presented. The implementation of a benefit system, the user-friendliness and the ecological information lead to a high attractiveness.

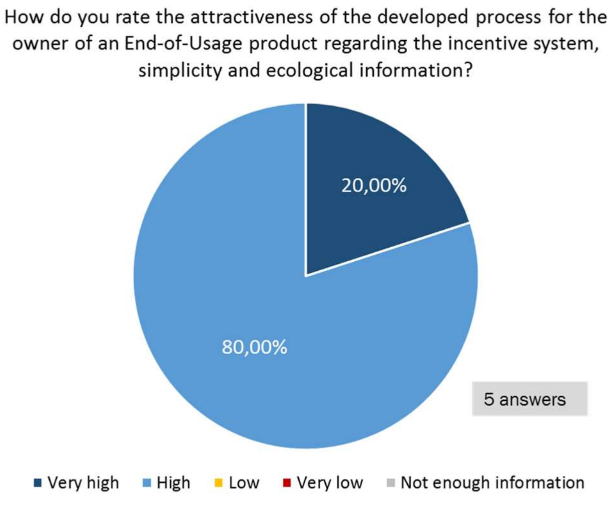


Figure 6-9: Attractiveness for the customer evaluated by experts

6.3.2 Scenario analysis

In order to check the potentials of the developed system, a short scenario analysis is carried out. It shows how many emissions can be saved after a successful implementation and training of the ANNs. The economic consideration is omitted here, since the profit margin is already defined in the economic evaluation. However, the analysis insists on assumptions made and is based only on numbers from Germany.

In order to determine the number of laptops that will reach the end of the usage phase in the future, the sales figures for the last years from Statista (2019) were used. The assumption is made that in Germany every year the same number of laptops reach their end of usage phase as have been sold on average over the last seven years. The determination of the proportions of disposed laptops, recycled laptops and reused/ remanufactured laptops is based on the study by Bitkom Research (2019). In the study, a distinction was made between storage, taking the laptop to a collection point for electronic waste, giving it away as a gift, donating it, taking it to a dealer and selling it. The stored laptops were distributed to all three groups in the scenarios. The laptops brought to the collection point are recycled and the remaining laptops are reused or remanufactured. Table 6-3 shows the assumed values and the emissions saved. Scenario A shows the current situation. Scenario B describes an improved situation.

Table 6-3: Scenario Analysis

	Scenario A	**Scenario B**
Disposed laptops [%]	7 %	2 %
Recycled laptops [%]	57 %	37 %
Reused/Remanufactured laptops [%]	36 %	61 %
Total number of laptops reaching end of usage phase	4.9 mio. pcs.	4.9 mio. pcs.
Saved emissions through more recycling		1,225 tonnes CO_2eq (245,000 laptops)
Saved emissions through more reuse/remanufacturing		202,615 tonnes CO_2eq (1,225,000 laptops)

Between Scenario A and Scenario B, the value of laptops disposed of has decreased from 7 % to 2 % and the value of reused or remanufactured laptops has increased to 61 %. It is assumed that every laptop that is recycled instead of disposed of saves 5 kg CO_2eq, which is equivalent to the recycling credit from Section 3.8. In order to determine the emissions saved through increased reuse and remanufacturing, it was assumed that on average 2.5 components per laptop have to be replaced. As a result, 75 % of the emissions from the manufacturing process are saved on average, which is 153.25 kg CO_2eq. The respective number of laptops then multiplies these values in order to determine the total quantity of saved emissions.

The sum of 203,840 tonnes CO_2eq is a considerable amount. However, if it is compared with the total emissions generated in Germany, which are 907 million tonnes CO_2eq in 2017 (Umwelt Bundesamt, 2017), there is a need to increase the values even more and to cover additional products.

6.4 SWOT analysis for South Africa

Since the development and current publications on CE often apply to the EU, the country-specific circumstances for South Africa are elaborated here. This is done in the form a SWOT analysis. This shows the opportunities, but also the threats, of the developed decision-making process. The strengths and weaknesses of South Africa are captured. Table 6-4 shows the results of the SWOT analysis.

Table 6-4: SWOT analysis for South Africa

Strengths • Many potential customers • Low cost of labour • Gateway to Africa with regard to research and technology • Quality of universities/ scientific and technical publications	**Weaknesses** • Volatile economy • Missing legal policies • Lower perception regarding sustainability • Lower focus on production engineering • Digital skills among population • Percentage of internet users
Opportunities • Many potential customer • 14th biggest polluter in the world • High unemployment (CE is labour intensive) • Cost reduction • Increase of prosperity	**Threats** • Significant mining activities • Transport of electronic waste from Europe

One of the strengths of South Africa is the high number of potential customers who would buy recovered products because they are cheaper (Marwala & Xing, 2011, p. 20). In addition, the labour costs are lower, which makes it attractive to carry out the manual activities there (Marwala & Xing, 2011, p. 20; World Economic Forum, 2018, p. 221). In addition, South

Africa is a gateway to Africa in terms of research and technology (Marwala & Xing, 2011, p. 20). South Africa performs very well in the quality of universities and technical publications (World Economic Forum, 2018, p. 221). This makes it possible in the future to implement the highly qualified tasks.

The volatile economy is considered to be a weakness of South Africa (Bag, Gupta, & Foropon, 2019). This complicates the implementation of a new system due to the associated uncertainties. Furthermore, there are no clear recycling regulations in South Africa (Marwala & Xing, 2011, p. 20). The use of legislation to drive change is not as prominent in South Africa as it is in the EU (Mativenga, Agwa-Ejon, Mbohwa, Sultan, & Shuaib, 2017, p. 286). Customers' demands for sustainability are not regarded as a driving factor (Mativenga et al., 2017, p. 286). There are few training opportunities offered that focus on production or manufacturing engineering (Marwala & Xing, 2011, p. 19). South Africa has a low level of digital skills among the population and a low percentage of internet users in a worldwide comparison (World Economic Forum, 2018, p. 221).

The high potential demand for recovered products is a great opportunity to implement the process (Marwala & Xing, 2011, p. 20). South Africa is the 14th largest producer of CO_2 emissions (Crippa et al., 2018). Germany takes sixth place. Moreover, South Africa has a high unemployment rate, which could be reduced by the labour-intensive CE (World Economic Forum, 2018, p. 220). By reusing resources, product components and products, costs can be saved, which benefits consumers. This is viewed, according to a survey by Mativenga et al. (2017, p. 286), as the main driver for recycling and reuse in South Africa. This leads to increased prosperity, because products can be bought which would otherwise be unaffordable, or more products can be bought (Marwala & Xing, 2011, p. 19).

Finally, the threats posed by the developed decision-making process are recorded. South Africa is a country with significant mining activities (Marwala & Xing, 2011, p. 19). CE tries to minimise the extraction of virgin materials (see Section 2.1.2). Therefore, when introducing the CE, care must be taken to ensure that the economy does not suffer as a result. Another threat is the transport of electronic waste from Europe to South Africa (Friege et al., 2015, p. 223). This could be achieved by labelling these shipments as products that can be reprocessed.

6.5 Chapter summary

This chapter serves to verify and validate the developed decision-making process. First, the V-model is presented, which is used as basis for this procedure. Then it is shown how the individual modules of the process have been verified during development and implementation. Subsequently, the entire process is validated on the basis of two surveys. The assessments of potential users and experts are analysed for this purpose. Finally a SWOT analysis for South Africa is carried out to take country-specific circumstances into account.

Chapter 7
Summary, conclusion and recommendations

This chapter presents a brief overview of the work that has been done in this thesis. The first section of this chapter summarises the content and results of the individual chapters. The next section describes how the research questions defined in Section 1.2 were answered and how this affected the further course of the research work. The results and findings are then critically discussed. The chapter concludes with the limitations and recommendations for further research.

7.1 Research summary

This research work is divided into seven chapters. Chapter 1 introduces the research project and describes the background of CE and rationale of the research topic. The problem statement and the research question derived from it are then presented as well as the defined research objectives and the research contribution. The applied research design and methodology are introduced and an outline of the thesis is shown.

Chapter 2 forms the literature review. It is divided into four sections and includes the CE, the decision-making process and ML with focus on ANN. These three sections are intended to provide insights and a basic theoretical knowledge, which are necessary to achieve the defined objectives. The fourth section shows the state of the art of ML regarding CE and decision-making.

The findings are used in Chapter 3 to first analyse the problems of the CE and to derive the requirements for an improved decision-making process. A model of the decision-making process is then developed based on these requirements. The study then examines which information can be used to determine the condition of EoU laptops and how it can be captured. The defined information and data structure is then used in the selection of the ML method. Afterwards a pairwise comparison between possible ML software alternatives is carried out on the basis of defined criteria in order to determine the most suitable one. Finally, the economic and ecological evaluation are developed.

Chapter 4 describes the simulation of comprehensive data, which will be used in the following chapter to demonstrate the developed decision-making process in practice and to train the ANNs. For this purpose, the independent variables which form the input for the ANN are first simulated on the basis of assumptions. The independent data is then used to determine the dependent data, which forms the output of the ANNs.

Based on the decision-making process developed theoretically in Chapter 3, it is implemented in chapter 5. First, the second part of the decision-making process, the determination of the condition of the laptop components, is modelled in Microsoft Azure ML studio. The structure and functionality of the ANNs is defined. The simulated data is then imported to train, test and evaluate the ANNs. Subsequently, the metrics achieved by the ANNs are discussed. After the determination of the condition of the laptop components has been implemented, the economic and ecological evaluation are programmed in Python.

Chapter 6 serves to verify and validate the developed decision-making process. First, the V-model is introduced, which structures the steps of this chapter. Since the verification of the individual development steps took place simultaneously with the development and implementation, it is shown here only in summary form. The entire process is then validated by conducting one survey with potential users and one survey with experts. In addition, a scenario analysis is carried out in order to show the potentials of the developed decision-making process. The chapter closes with a SWOT analysis for South Africa in order to consider country-specific circumstances.

7.2 Research conclusions and contribution

The results and conclusions are structured according to the research questions defined at the beginning of the study. First, the key findings are summarised in order to answer the secondary research questions. These lead to the primary research question, which is then answered.

Secondary research question 1: *"What are the problems related to and causes of the low return rates in the CE with regard to the decision-making process?"*

The answer contains several factors that were identified in the literature review (see Chapter 2). High labour costs combined with a low degree of automation and unnecessary manual quality checks of the EoU products make the recovery economically unattractive. In addition, there is unnecessary transportation of EoU products that cannot be recovered profitably and therefore have to be sent back to the owner. Shortened life cycles and the increasing diversity of modern products make it difficult to generate the required know-how for the workforce. Moreover, recovery is associated with uncertainties regarding quality and quantity, which leads to economic uncertainties that prevent investment. Therefore, often only functional devices are purchased and the rest are recycled or disposed of, although it might have been possible to remanufacture them. This is partly due to the fact that many of the products currently in use have not been developed for the CE. This makes condition monitoring or determination of their condition difficult. The same applies to the replacement of components if they cannot be reused directly, as some types of construction do not allow this. In addition, current incentive schemes only apply to functional products and not for defective products.

Secondary research question 2: *"What are the requirements for an improvement of the decision-making process regarding the functions and information required?"*

These requirements are derived from the problems identified previously and serve to develop a revised decision-making process (see Sections 3.1 and 3.2). It should be possible to make a decentralised preselection and should be attractive for both the user and the company offering the service. The information required must be easy to determine for the owner of the laptop. Furthermore, it should be possible to record them without upgrading the laptop or a specific development for the CE. Therefore, the process is divided into two parts. First, a GUI is used to enter information from the owner about the EoU laptop that can be used to determine the condition of defined components. The determined quality of the EoU product is then evaluated economically and ecologically in order to select the best of the possible recovery options. If the laptop can be recovered economically, the possible purchase price is communicated to the owner, otherwise it is recommended to recycle the laptop and a price for the resource is communicated.

Secondary research question 3: *"How can the condition of components of laptops not designed for the CE and not physically present be determined?"*

The condition of components depends on the initial quality and the impact of stress factors. A distinction can be made between environmental stress and operational stress. In addition, the possible points in time when the stress factors occurred must be determined. The brand and model can be determined at any time. The impact of stress factors on usage must be continuously recorded during the usage phase to accurately describe them. Since the products were not designed for the purpose of CE, alternative information must be used to determine how strongly they have been influenced by the stress factors. Therefore, the age, the type of use, mobile and the use of a protective case are defined in order to provide information about the usage time and intensity of usage of individual components. This is extended by external and obvious damage to certain parts of the laptop in order to obtain further information on stress factors. Section 3.3 and 3.4 elaborate this information.

Secondary research question 4: *"Which ML method is suitable for determining the condition of the laptop components based on the defined information and which software can best implement this?"*

The structure of the defined decision-making process and the associated information require a classification process. The output classes are defined and there are complete training datasets available, which include input and the corresponding output. Therefore, a learning algorithm from the group of supervised learning is selected. This is where feedforward ANNs come in, as they can process a wide variety of data types and uncover complicated correlations in the data. Since only two values are possible as outputs in the determination of the condition of a component and only one of the values can be assigned to a component, a two-class single label classification must be applied. This means that the ANN can only deliver one label per run as output and is only trained on two different classes. This implies that one ANN per component must be developed and trained, whereas these are very similar in structure and functionality. ANNs make high demands on the computing power of the hardware. However, this is not a problem due to the central execution of the calculation processes in the cloud, since computing capacities can be made available there very easily and flexibly.

Secondary research question 5: *"How can an EoU laptop recovery be economically and ecologically evaluated?"*

The developed decision-making process defines that the evaluation of the laptop recovery, which forms the second part of the process, is based on the previously determined condition of the laptop components. Section 2.2.3 elaborates on decision factors that can be divided into economic, environmental and social factors. The economic evaluation developed in Section 3.7 is based primarily on the possible selling price of the recovered laptop, which is defined by an expert or a prognosis system and the condition of the laptop. The calculated costs for the recovery will be offset against the selling price to determine the possible purchase price of the EoU laptop. These are calculated on the basis of the components to be replaced. This also applies to the ecological evaluation. The ecological evaluation determines the emissions saved by the recovery of the laptop instead of disposing it. For this purpose, the part of the PCF of the manufacturing and the conditions of the components are used. Components that need to be replaced reduce the emissions saved, as their manufacture and distribution causes emissions. The decision-making process developed here depends only on the result of the economic evaluation and the ecological evaluation serves only as additional information for the customer. However, this can be actively incorporated into further development of the system. If the economic evaluation comes to the conclusion that the laptop can only be recycled, the laptop can still be bought for the value of the resources it contained.

Primary research question: *"How can the decision-making process of EoU products/ laptops be improved in order to increase the return rates?"*

The answers to the secondary research questions are incorporated in the answer to the primary research question. The decision-making process regarding the subsequent steps of an EoU product/ laptop in the CE can be improved by a decentralised execution by the owner. This is made possible by the use of ML. Existing, easy-to-capture information is used to determine the quality of the laptop at component level. These must be able to be determined even if the laptop was not designed for the CE and no special upgrade was carried out before the usage-phase, as installation of software or sensors. Based on this, an economic and ecological evaluation can be carried out to determine the next step and inform the owner about the best option. Increased attractiveness and reduced uncertainties encourage investment by companies and use by customers. In addition, costs are saved through the use of ML, as unnecessary transport and dismantling are avoided. Furthermore, the ANNs can learn from all objects recorded in the system. This leads to a faster generation of know-how and makes the process more flexible. The information regarding the environmental aspects increases the motivation of EoU laptop owners who value sustainability and strengthens the awareness of people who have not yet addressed the issue. To use the process for products other than laptops, adjustments must be made to the required information and regarding components.

7.3 Limitations and recommendations for further research

First, the limitation that the training data of the ANNs are simulated must be taken into account. Although correlations between variables are incorporated on the basis of research findings and noise was also implemented, they do not represent data generated and captured in reality. The implemented values for the residual value of resources, emissions per component and recovery costs are also based on assumptions or general values and are not individually determined for the real situation. The simulation of the data also leads to the availability of a comprehensive quantity of data, which is not the case in reality at the outset.

The next limitation of the developed decision-making process is the differentiation between repair and remanufacturing. Not only is a distinction made between defective and non-defective components, but components, which only have a short remaining service life, are also exchanged. The ANNs can therefore be trained either for repair or for remanufacturing. This is done by defining which components are marked in the training data that they have to be exchanged. Furthermore, in the case of the recycling proposal, not only the residual value of the resources should be calculated, but also the value should be calculated which can be generated by cannibalising the device, based on the predetermined condition of the components.

A further limitation is that the determination process of the laptop components developed in this research is specially adapted for devices, which have not been developed for the CE. For products and devices that are equipped with sensors and software to monitor and record the stress factors or the condition of components, the process must be extended to include this information, as this would significantly improve the result.

Further research should therefore be conducted based on data collected from real laptops. This could lead to insights that require a specific adaptation of the decision-making process developed in this research work. The improvement in input information variables should also be considered. In the course of the research with real data, ANNs should be defined as small as possible but as complex as necessary. The quality of the results must be satisfactory in order to be able to make reliable statements and predictions. However, the amount of training data required is a strong cost driver and should therefore be as low as possible. Moreover, other ML methods could be tested with the new data.

The developed process could be improved by an order function. A customer would choose a certain laptop model and determine which components should be upgraded, e.g. a better CPU is required. For this, a laptop could be used, where exactly this component is defective in order to save costs, emissions and resources. Another point is the combination of economic and ecological factors. So far, it is almost not possible to value both of them in one key figure and so one of the two must be chosen as the decision maker.

A particularly large potential for improving the quality determination of an EoU laptop is the use of data which the laptop can already generate, but which is not recorded without the right software. This includes the temperatures of individual components that are only used to control the cooling system. In addition, the utilisation of CPU and GPU, as well as number of charging cycles and actual usage time are very meaningful parameters. An application that records this information and analyses them as the developed process, combined with self-tests of individual components, as is already available for smartphone batteries, offers great potential.

Reference list

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*(11).

Agrawal, V. V., Atasu, A., & van Ittersum, K. (2015). Remanufacturing, Third-Party Competition, and Consumers' Perceived Value of New Products. *Management Science, 61*(1).

Alpar, P., Alt, R., Bensberg, F., Grob, H. L., Weimann, P., & Winter, R. (2014). Phasenmodelle in der Systementwicklung. In P. Alpar, R. Alt, F. Bensberg, H. L. Grob, P. Weimann, & R. Winter (Eds.), *Anwendungsorientierte Wirtschaftsinformatik* (pp. 315–359). Wiesbaden: Springer Fachmedien Wiesbaden.

Alsmadi, M. K. S., Bin Omar, K., & Noah, S. A. (2009). Back Propagation Algorithm: The Best Algorithm Among the Multi-layer Perceptron Algorithm. *International Journal of Computer Sience and Network Security, 9*(4).

Alves Dias, P., Ardente, F., Blagoeva, D., Blengini, G. A., Bobba, S., Buraoui, F., et al. (2018). *Report on critical raw materials and the circular economy.* Luxembourg: Publications Office of the European Union.

Amezquita, T., Hammond, R., Salazar, M., & Bras, B. (1995). Characterizing the remanufacturability of engineering systems. *Proceedings 1995 ASME Advances in Design Automation Conference.* (82), 271–278.

Andrew-Munot, M., N Ibrahim, R., & Junaidi, E. (2015). An Overview of Used-Products Remanufacturing. *Mechanical Engineering Research, 5*(1), 12–23.

Angelis, R. de (2018). *Business Models in the Circular Economy: Concepts, Examples and Theory.* Cham: Springer International Publishing.

Avramenko, Y., & Kraslawski, A. (2008). *Case Based Design: Applications in Process Engineering. Studies in Computational Intelligence: Vol. 87.* Berlin, Heidelberg: Springer-Verlag.

Backhaus, K., Erichson, B., Plinke, W., & Weiber, R. (2016). *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung* (14., überarbeitete und aktualisierte Auflage). Berlin, Heidelberg: Springer Gabler.

Backhaus, K., Erichson, B., & Weiber, R. (2015). *Fortgeschrittene Multivariate Analysemethoden: Eine anwendungsorientierte Einführung* (3., überarbeitete und aktualisierte Auflage). Berlin, Heidelberg: Springer Gabler.

Bag, S., Gupta, S., & Foropon, C. (2019). Examining the role of dynamic remanufacturing capability on supply chain resilience in circular economy. *Management Decision, 57*(4), 863–885.

Baker, D., Bridges, D., Hunter, R., Johnson, G., Krupa, J., Murphy, J., & Sorenson, K. B. (2001). *Guidebook to Decision-Making Methods.* WSRC-IM-2002-00002, Department of Energy.

Baldé, C.p., Forti, V., Gray, V., Kuehr, R., & Stegmann, P. (2017). *The Global E-waste Monitor: Quantities, Flows, and Resources.* Bonn, Geneva, Vienna: United Nations University (UNU), International Telecommunication Union (ITU) & International Solid Waste Association (ISWA).

Barquet, A. P., Rozenfeld, H., & Forcellini, F. A. (2013). An integrated approach to remanufacturing: model of a remanufacturing system. *Journal of Remanufacturing, 3*(1).

Bastein, T., Roelofs, E., Rietveld, E., & Hoogendoorn, A. *Opportunities for a circular economy in the netherlands:* TNO.

Becker, H. (2015). *Wie lange hält ein Laptop?: Verschleiß und Haltbarkeit.* Retrieved 17. Novemeber 2019, from https://praxistipps.chip.de/wie-lange-haelt-ein-laptop-verschleiss-und-haltbarkeit_42505.

Bektas, O., Jones, J. A., Sankararaman, S., Roychoudhury, I., & Goebel, K. (2018). A neural network filtering approach for similarity-based remaining useful life estimation. *The International Journal of Advanced Manufacturing Technology, 101*(1-4), 87–103.

Bhargava, N., Aziz, A., & Arya, R. (2013). Slection Criteria for Data Mining Software: A Study. *International Journal of Computer Siences Issues, 10*(3), 308–312.

Bitkom Research (2019). *Alte Laptops in deutschen Haushalten.* Retrieved November 17, 2019, from Digitalverband Bitkom: https://www.bitkom.org/Presse/Presseinformation/Jeder-Zweite-besitzt-einen-alten-Laptop-oder-PC.

Blume, V. (2003). *Informations- und Entscheidungsmanagement in der Kreislaufwirtschaft: Anforderungen zur Ausgestaltung des Informationsmanagements für die unternehmensübergreifende Entscheidungsunterstützung in Kreislaufwirtschaftssystemen. Europäische Hochschulschriften Reihe 5, Volks- und Betriebswirtschaft: Vol. 2965.* Frankfurt am Main: Lang.

Boehm, B. W. (1979). Guidelines for verifying and validating software requirements and design specifications. *Euro IFIP 79*.

Box, G. E. P. (1976). Science and Statistics. *Journal of the American Statistical Association, 71*(356), 791–799.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159.

Braun, A. T., Kleine-Moellhoff, P., Reichenberger, V., & Seiter, S. (2018). Case Study Analysing Potentials to Improve Material Efficiency in Manufacturing Supply Chains, Considering Circular Economy Aspects. *Sustainability, 10*(3), 880.

Braun, A. T., Mandel, J., & Bauernhansl, T. (2013). Effective Use of Resources in Closed Value Networks. In T. Bossomaier (Ed.), *Proceedings: 25 - 27. September 2013. SESDE 2013, 1st International Workshop on Simulation for Energy, Sustainable Development & Environment* (pp. 12–15). Athen, Greece.

Braungart, M., McDonough, W., & Bollinger, A. (2006). Cradle-to-cradle design: creating healthy emissions – a strategy for eco-effective product and system design. *Journal of Cleaner Production, 15*(13-14), 1337–1348.

Celaya, J. R., Wysocki, P., Vashchenko, V., Saha, S., & Goebel, K. (2010). Accelerated Aging System for Prognostics of Power Semiconductor Devices. *IEEE*.

Crippa, M., Guizzardi, D., Muntean, M., Olivier, J.G.J., Schaaf, E., Solazzo, E., & Vignati, E. (2018). *Fossil CO2 emissions of all world countries: 2018 report. EUR, Scientific and technical research series: Vol. 29433.* Luxembourg: Publications Office of the European Union.

Deng, W. J., Louie, P.K.K., Liu, W. K., Bi, X. H., Fu, J. M., & Wong, M. H. (2006). Atmospheric levels and cytotoxicity of PAHs and heavy metals in TSP and PM2.5 at an electronic waste recycling site in southeast China. *Atmospheric Environment, 40*(36), 6945–6955.

Dunmade, I. (2004). PLETS model: a sustainability-concept-based approach to product end-of-life management. *Proceedings Volume 5583, Environmentally Conscious Manufacturing IV*, 118–126.

Dyckhoff, H., Souren, R., & Keilen, J. (2004). The Expansion of Supply Chains to Closed Loop Systems: A Conceptual Framework and the Automotive Industry's Point of View. In H. Dyckhoff, R. Lackes, & J. Reese (Eds.), *Supply chain management and reverse logistics* (1st ed.). Berlin: Springer.

Dyllick, T., & Hockerts, K. (2002). Beyond the business case for corporate sustainability. *Business Strategy and the Environment, 11*(2), 130–141.

El Jerjawi, N. S., & Abu-Naser, S. S. (2018). Diabetes Prediction Using Neural Network. *International Journal of Advanced Sience and Technology*. (121), 54–64.

Ellen MacArthur Foundation (2013). *Towards the circular economy: Economic and business rationale for an accelerated transition.* Retrieved October 01, 2019, from www.ellenmacarthurfoundation.org.

Ellen MacArthur Foundation (2015a). *Growth within: A circular economy vision for a competitive europe.*

Ellen MacArthur Foundation (2015b). *Towards a Circular Economy: Business Rationale for an Accelerated Transition.*

Ellen MacArthur Foundation (2018). *Circular Consumer Electronics: An initial exploration.* Retrieved October 01, 2019, from www.ellenmacarthurfoundation.org.

Ellen MacArthur Foundation (2019). *Artificial Intelligence and the Circular Economy: AI as a Tool to Accelerate the Transition.*

Engel, H., Stuchtey, M., & Vanthournout, H. (2016). Ahead of the curve: Innovative models for waste management in emerging markets: How countries can turn their garbage into wealth. In Murphy Cait & J. Rosenfield (Eds.), *The circular economy. Moving from theory to practice.* McKinsey Company.

European Commission (2017). *Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of regions on the 2017 list of Critical Raw Materials for the EU.* Brussels: European Commission.

European Environment Agency (2014). *Resource-efficient green economy and EU policies. EEA report: no 2/2014.* Copenhagen: European Environment Agency.

European Parliament and the Council of the European Union (2008). Directive 2008/98/EC of the European Parliament and of the Council of 10 Novemeber 2008 on waste and repealing certain Directives. *Official Journal of the European Union*. (51).

European Parliament and the Council of the European Union (2012). Directive 20129C of the European Parilament and the Council of 4 July 2012 on waste electrical and electronic equipment (WEEE). *Official Journal of the European Union*. (197).

Evgeniou, T., & Pontil, M. (2001). Support Vector Machines: Theory and Applications. In G. Goos, J. Hartmanis, J. van Leeuwen, G. Paliouras, V. Karkaletsis, & C. D. Spyropoulos (Eds.), *Lecture Notes in Computer Science. Machine Learning and Its Applications* (pp. 249–257). Berlin, Heidelberg: Springer Berlin Heidelberg.

Fahrmeir, L., Heumann, C., Künstler, R., Pigeot, I., & Tutz, G. (2016). *Statistik: Der Weg zur Datenanalyse* (8., überarbeitete und ergänzte Auflage). *Springer-Lehrbuch.* Berlin, Heidelberg: Springer Spektrum.

Ferguson, M. E., Fleischmann, M., & Souza, G. C. (2011). A Profit-Maximizing Approach to Disposition Decisions for Product Returns. *Decision Sciences, 42*(3), 773–798.

Friege, H. (2012). Review of material recovery from used electric and electronic equipment-alternative options for resource conservation. *Waste management & research : the journal of the International Solid Wastes and Public Cleansing Association, ISWA, 30*(9), 3–16.

Friege, H., Oberdörfer, M., & Günther, M. (2015). Optimising waste from electric and electronic equipment collection systems: a comparison of approaches in European countries. *Waste management & research : the journal of the International Solid Wastes and Public Cleansing Association, ISWA, 33*(3), 223–231.

Gartner, I. (2018). *5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018.* Retrieved November 17, 2019, from https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/.

Ghazalli, Z., & Murata, A. (2011). Development of an AHP–CBR evaluation system for remanufacturing: end-of-life selection strategy. *International Journal of Sustainable Engineering, 4*(1), 2–15.

Goncalves, L., Subtil, A., Oliveira, M. R., & De Zea Bermudez, Patricia de (2014). Roc Curve Estimation: An Overview. *Revstat - Statistical Journal*. (1), 1–20.

Goodall, P. A. (2014). *Decision support for assessing the feasibility of a product for remanufacture.*

Goodall, P. A., Rosamond, E. L., & Harding, J. A. (2014). A review of the state of the art in tools and techniques used to evaluate remanufacturing feasibility. *Journal of Cleaner Production*. (81), 1–15.

Goodall, P., Graham, I., Harding, J., Conway, P., Schleyer, S., & West, A. (2015). Cost estimation for remanufacture with limited and uncertain information using case based reasoning. *Journal of Remanufacturing, 5*(1).

Grossi, V., Pedreschi, D., & Turini, F. (2016). Data Mining and Constraints: An Overview. In C. Bessiere, L. de Raedt, L. Kotthoff, S. Nijssen, B. O'Sullivan, & D. Pedreschi (Eds.), *Lecture Notes in Computer Science. Data Mining and Constraint Programming* (pp. 25–48). Cham: Springer International Publishing.

Guide Jr., V. Daniel R. (2000). Production planning and control for remanufacturing: industry practice and research needs. *Journal of Operations Management*. (18), 467–483.

Guo, K. L. (2008). DECIDE: a decision-making model for more effective decision making by health care managers. *The health care manager, 27*(2), 118–127.

Gurita, N., Fröhling, M., & Bongaerts, J. (2018). Assessing potentials for mobile/smartphone reuse/remanufacture and recycling in Germany for a closed loop of secondary precious and critical metals. *Journal of Remanufacturing, 8*(1), 1–22.

Gutowski, T. G., Sahni, S., Boustani, A., & Graves, S. C. (2011). Remanufacturing and energy savings. *Environmental science & technology, 45*(10), 4540–4547.

Haberer, R. (2009). *Langzeitstudie: Jedes dritte Notebook defekt.* Retrieved November 17, 2019, from https://www.netzwelt.de/news/81224-langzeitstudie-dritte-notebook-defekt.html.

Han, X., Wu, H., Yang, Q., & Shang, J. (2016). Reverse channel selection under remanufacturing risks: Balancing profitability and robustness. *International Journal of Production Economics, 182*, 63–72.

Hannon, E., Kuhlmann, M., & Thaidigsmann, B. (2016a). Developing products for a circular economy: Cross-functional collaboration and customer-focused design thinking can help companies reap more value from energy and resources they use. In Murphy Cait & J. Rosenfield (Eds.), *The circular economy. Moving from theory to practice.* McKinsey Company.

Hannon, E., Magnin-Mallez, C., & Vanthournout, H. (2016b). Introduction. In Murphy Cait & J. Rosenfield (Eds.), *The circular economy. Moving from theory to practice* (pp. 2–3). McKinsey Company.

Haykin, S. S. (2009). *Neural networks and learning machines* (3. ed.). New York: Pearson.

Heck, P. (2006). *Circular Economy related international practices and policy trends:: Current situation and practices on sustainable production ans consumption and international Circular Economy:* Institut für angewandtes Stoffstrommanagement.

Hering, E. (1984). Programmablaufplan nach DIN 66001. In E. Hering (Ed.), *Software-Engineering* (pp. 26–34). Wiesbaden: Vieweg+Teubner Verlag.

Hildebrand, D. (2005). *Economic analyses of vertical agreements - a self-assessment. International competition law series: Vol. 17.* The Hague: Kluwer Law Internat.

Hübner, R. (2013). *Planned Obsolence. Geplante Obsolenz - ein umstrittenes Phänomen.*

Hübner, R., Himpelmann, M., Melnitzky, S., Stahel, W. R., & Hübner, H. (2006). Reprocessing gebrauchter Güter: Eine Strategie der Nachhaltigkeit und ihre Auswirkungen auf die Lieferketten einer "Fabrik der Zukunft". *Berichte aus Energie- und Umweltforschung, 33*.

Inderfurth, K. (2004). Product Recovery Behaviour in a Closed Loop Supply Chain. In H. Dyckhoff, R. Lackes, & J. Reese (Eds.), *Supply chain management and reverse logistics* (1st ed.). Berlin: Springer.

International Resource Panel (2017). *Assessing Global Resource Use.* Nairobi, Kenya: United Nations Environement.

Jadhav, A. S., & Sonar, R. M. (2011). Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software, 84*(8), 1394–1407.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning: With applications in R* (Corrected at 8th printing). *Springer texts in statistics.* New York, Heidelberg, Dordrecht, London: Springer.

Jun, H.-B., Cusin, M., Kiritsis, D., & Xirouchakis, P. (2007). A multi-objective evolutionary algorithm for EOL product recovery optimization: turbocharger case study. *International Journal of Production Research, 45*(18-19), 4573–4594.

Jun, H.-B., Lee, D.-H., Kim, J.-G., & Kiritsis, D. (2012). Heuristic algorithms for minimising total recovery cost of end-of-life products under quality constraints. *International Journal of Production Research, 50*(19), 5330–5347.

Kandel, E. R., Schwartz, J. H., Jessell, T., Siegelbaum, S. A., Hudspeth, A. J., & Mack, S. (Eds.) (2013). *Principles of neural science* (Fifth edition). New York, Lisbon, London: McGraw-Hill Medical.

Kaza, S., Yao, L., Bhada-Tata, P., & van Woerden, F. (2018). *What a Waste 2.0: A Global Snapshot of Solid Waste Management To 2050. Urban Development Series.* Washington, D. C: World Bank Publications.

Klein, M. (2010). Add-In basierte Softwaretools zur stochastischen Unternehmensbewertung? Spreadsheet basierte Monte-Carlo-Simulation und Risikoanalyse bei den vier marktführenden Softwarepaketen im Vergleich. *Working Papers in Acoounting Valuation Auditing*. (7).

Kneuper, R. (2018). *Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes.* Cham: Springer International Publishing.

Kraak, D., Taouil, M., Hamdioui, S., Weckx, P., Catthoor, F., Chatterjee, A., et al. (2018). Device aging: A reliability and security concern. In *2018 IEEE 23rd European Test Symposium (ETS)* (pp. 1–10). IEEE.

Kruse, R., Borgelt, C., Braune, C., Klawonn, F., Moewes, C., & Steinbrecher, M. (2015). *Computational Intelligence: Eine methodische Einführung in künstliche neuronale Netze, evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze* (2., überarbeitete und erweiterte Auflage). *Computational intelligence.* Wiesbaden: Springer Vieweg.

Kumar, S., Sotiris, V., & Pecht, M. (2008). Mahalanobis Distance and Projection Pursuit Analysis for Health Assessment of Electronic Systems. *IEEE Aerospace Conference*, 3674–3683.

Kuo, W., Chien, W.-T. K., & Kim, T. (1998). Reliability Stress Tests. In W. Kuo, W.-T. K. Chien, & T. Kim (Eds.), *Reliability, Yield, and Stress Burn-In* (pp. 93–123). Boston, MA: Springer US.

Lange, U. (2017). Ressourceneffizienz durch Remanufacturing - Industrielle Aufarbeitung von Altteilen: Kurzanalyse Nr. 18. *VDI ZRE Publikationen*.

Lieder, M., Asif, F. M. A., Rashid, A., Mihelič, A., & Kotnik, S. (2017). Towards circular economy implementation in manufacturing systems using a multi-method simulation approach to link design and business strategy. *The International Journal of Advanced Manufacturing Technology, 93*(5-8), 1953–1970.

Long, E., Kokke, S., Lundie, D., Shaw, N., Ijomah, W., & Kao, C.-C. (2016). Technical solutions to improve global sustainable management of waste electrical and electronic equipment (WEEE) in the EU and China. *Journal of Remanufacturing, 6*(1), 243.

Macauley, M., Palmer, K., & Shih, J.-S. (2003). Dealing with electronic waste: modeling the costs and environmental benefits of computer monitor disposal. *Journal of Environmental Management, 68*(1), 13–22.

Manoj krishna, M., Neelima, M., Harshali, M., & Venu Gopala Rao, M. (2018). Image classification using Deep learning. *International Journal of Engineering & Technology, 7*(2.7), 614.

Marin, C., Alféres, G. H., Córdova, J., & González, V. (2015). Detection of melanoma through image recognition and artificial networks. *World Congress on Medical Physics and Biomedical Engineering, Toronto, Canada*.

Marwala, T., & Xing, B. (2011). The role of remanufacturing in building a developmental state. *The Thinker: For the Thought Leaders*, 18–20.

Mativenga, P. T., Agwa-Ejon, J., Mbohwa, C., Sultan, A. A. M., & Shuaib, N. A. (2017). Circular Economy Ownership Models: A view from South Africa Industry. *Procedia Manufacturing, 8*, 284–291.

Matsumoto, M., & Ijomah, W. (2013). Remanufacturing. In J. Kauffman & K.-M. Lee (Eds.), *Handbook of Sustainable Engineering* (pp. 389–408). Dordrecht: Springer Netherlands.

McKinsey Global Institute (2011). *Resource Revolution: Meeting the world's energy, materials, food, and water needs.* Retrieved October 01, 2019, from https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/Sustainability/Our%20Insights/Resource%20revolution/MGI_Resource_revolution_full_report.ashx.

Mentis, M. (2015). Managing project risks and uncertainties. *Forest Ecosystems, 2*(1), 1–13.

Moret, B. M. E. (1982). Decision Trees and Diagrams. *ACM Computing Surveys, 14*(4), 593–623.

Morgan, J., & Mitchell, P. (2015). *Employment and the circular economy: Job creation in a more resource efficient Britain.* London: Green Alliance.

Mukherjee, K., Mondal, S., & Chakraborty, K. (2017). Impact of various issues on extending the useful life of a product through product recovery options. *Journal of Remanufacturing, 7*(1), 77–95.

O'Connell, S., & Stutz, M. (2010). Product carbon footprint (PCF) assessment of Dell laptop - Results and recommendations. *Proceedings of the 2010 IEEE International Symposium on Sustainable Systems and Technology*, 1–6.

OECD (2019). *Global material resources outlook to 2060: Economic drivers and environmental consequences.* Paris: OECD Publishing.

Östlin, J. (2008). *On Remanufacturing Systems: Analysing and Managing Material Flows and Remanufacturing Processes. Linköping studies in science and technology. Thesis: Vol. 1192.* Linköping: Department of Management and Engineering, Linköpings universitet.

Panasonic (2012). *Reliability of Semiconductor Devices.* Retrieved October 01, 2019, from https://pdfs.semanticscholar.org/9f4a/695d29ef7ab784ad0dfa448fb9bd8feeaaf8.pdf.

Parajuly, K., & Wenzel, H. (2017). Potential for circular economy in household WEEE management. *Journal of Cleaner Production, 151*, 272–285.

Paterson, D. A. P., Kao, C.-C., Ijomah, W. L., & Windmill, J. F. C. (2018). Incorporating remanufacturing into the end-of-life vehicles directive: current presence and the waste problem. *Journal of Remanufacturing*. (8), 23–37.

Patona International (2018). *Wann sollte der Laptop Akku ersetzt werden?* Retrieved November 17, 2019, from https://www.pts-trading.de/news/wann-akku-ersetzen/.

Patona International (2019). *Die maximale Lebensdauer für Ihren Laptop Akku.* Retrieved November 17, 2019, from https://www.pts-trading.de/news/maximale-lebensdauer-laptop-akku/.

Pearce, D. W., & Turner, R. K. (1990). *Economics of Natural Resources and the Environment.* Hemel Hemstead: Harvester Wheatsheaf.

Pinegger, M. (2018). *Neuronale Netze. Einsatzmöglichkeiten künstlicher Intelligenz in der Kreislaufwirtschaft* (1. Auflage). München: ScienceFactory.

Pineyro, P., & Viera, O. (2018). Heuristic procedure for the economic lot-sizing problem with remanufacturing and recovery targets. *Journal of Remanufacturing*.

Prakash, S., Antony, F., Köhler, A. R., & Liu, R. (2016). *Ökologische und ökonomische Aspekte beim Vergleich von Arbeitsplatzcomputern für den Einsatz in Behörden unter Einbeziehung des Nutzerverhaltens (Öko-APC).* Dessau-Roßlau: Umweltbundesamt.

Prakash, S., Liu, R., Schischke, K., & Stobbe, L. (2012). *Zeitlich optimierter Ersatz eines Notebooks unter ökologischen Gesichtspunkten.* Dessau-Roßlau: Umweltbundesamt.

Presley, A., Meade, L., & Sarkis, J. (2007). A strategic sustainability justification methodology for organizational decisions: a reverse logistics illustration. *International Journal of Production Research, 45*(18-19), 4595–4620.

Preston, F. (2012). A Global Redesign? Shaping the Circular Economy. *EERG BP, 2012*(02).

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Ramadoss, T. S., Alam, H., & Seeram, R. (2018). Artificial Intelligence and Internet of Things enabled Circular economy. *The International Journal of Engineering and Science, 7*(9), 55-63.

Richert, W., & Coelho, L. P. (2013). *Building machine learning systems with Python.* Birmingham, UK: Packt Publishing.

Rizos, V., Tuokko, K., & Behrens, A. (2017). *The circular economy, a review of definitions, processes and impacts. Research report / CEPS: No 2017/08.* Brussels: CEPS.

Rossé, M., Stuchtey, M., & Vanthournout, H. (2016). Finding growth within: A new framework for Europe: How the circular economy could promote clean growth. In Murphy Cait & J. Rosenfield (Eds.), *The circular economy. Moving from theory to practice.* McKinsey Company.

Royce, W. (1970). Managing the development of large software systems. *Proceedings, IEEE WESCON*, 328–338.

Rumpe, B. (2016). *Modeling with UML: Language, Concepts, Methods.* Cham: Springer International Publishing.

Sathya, R., & Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence, 2*(2).

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for business students* (5. ed.). Harlow: Financial Times Prentice Hall.

Sauvé, S., Bernard, S., & Sloan, P. (2016). Environmental sciences, sustainable development and circular economy: Alternative concepts for trans-disciplinary research. *Environmental Development, 17*, 48–56.

Schmallowsky, K., & Reimers, H.-E. (Eds.) (2015). *WDP - Wismarer Diskussionspapiere / Wismar Discussion Papers: Vol. 2015,02. Unternehmensbewertung mit Monte-Carlo-Simulationen.* Wismar: Fakultät für Wirtschaftswissenschaften Wismar Business School.

Seitz, M. A. (2007). A critical assessment of motives for product recovery: the case of engine remanufacturing. *Journal of Cleaner Production, 15*(11-12), 1147–1157.

Sheehan, E., Braun, A.-T., Kuhlmann, T., & Sauer, A. (2016). Improving Material Efficiency for Ultra-efficient Factories in Closed-loop Value Networks. *Procedia CIRP, 40*, 455–462.

Simchi-Levi, D., Wu, S. D., & Shen, Z.-J. (Eds.) (2004). *International Series in Operations Research & Management Science: Vol. 74. Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era.* Boston, MA, s.l.: Springer US.

Simeone, O. (2018). *A Very Brief Introduction to Machine Learning With Applications to Communication Systems.* Retrieved November 17, 2019, from http://arxiv.org/pdf/1808.02342v4.

Singh, H., & Jain, P. K. (2016). Remanufacturing of functional surfaces using developed ECH machine. *Journal of Remanufacturing, 6*(1), 77.

Squaretrade (2009). *1 in 3 Laptops fail over 3 years.* Retrieved November 17, 2019, from https://www.squaretrade.com/htm/pdf/SquareTrade_laptop_reliability_1109.pdf.

Stadt Stuttgart (2019). *Diesel Verkehrsverbot.* Retrieved November 17, 2019, from https://www.stuttgart.de/diesel-verkehrsverbot.

Statista (2019). *Absatz von Desktop-PCs und Notebooks in Deutschland bis 2018.* Retrieved November 02, 2019, from https://de.statista.com/statistik/daten/studie/28292/umfrage/absatzzahlen-fuer-desktop-pcs-und-notebooks-seit-2005/.

Statistisches Bundesamt (Destatis) (2018). *Produzierendes Gewerbe: Kostenstruktur der Unternehmen des Verarbeitenden Gewerbes sowie des Bergbaus und der Gewinnung von Steinen und Erden.*

Sundin, E., & Bras, B. (2005). Making functional sales environmentally and economically beneficial through product remanufacturing. *Journal of Cleaner Production, 13*(9), 913–925.

Sundin, E., Sakao, T., Lindahl, M., Kao, C.-C., & Joungerious, B. (2016). *Map of Remanufacturing Business Model Landscape.*

Thierry, M., Salomon, M., van Nunen, J., & van Wassenhove, L. (1995). Strategic Issues in Product Recovery Management. *California Management Review*. (2), 114–135.

Tilton, J. (2013). Rolls-Royce Corporation reduces cost and improves efficiency in the Reverse Logistics & Remanufacturing Supply Chain by applying LEAN 6-Sigma methods. *Reverse Logistics Magazine, 58*, 12–22.

Uckun, S., Goebel, K., & Lucas, P. J.F. (2008). Standardizing research methods for prognostics. *International Conference on Prognostics and Health Management*, 1–10.

Umwelt Bundesamt (2017). *Treibhausgas-Emissionen in Deutschland.* Retrieved November 17, 2019, from https://www.umweltbundesamt.de/daten/klima/treibhausgas-emissionen-in-deutschland#textpart-1Umwelt Bundesamt.

Unterstein, M., & Matthiessen, G. (2012). *Relationale Datenbanken und SQL in Theorie und Praxis.* Berlin, Heidelberg: Springer Berlin Heidelberg.

van Nunen, J. A.E.E., & Zuidwijk, R. A. (2004). E-Enabled Closed-Loop Supply Chains. *California Management Review, 46*(2), 40–54.

World Economic Forum (2018). *Readiness for the future of production report 2018. Insight report.* Cologny/Geneva: World Economic Forum.

World Economic Forum (2019). *A New Circular Vision for Electronics: Time for a Global Reboot.* Switzerland: The Platform for Accelerating the Circular Economy.

Xu, Y., Fernandez Sanchez, J., & Njuguna, J. (2014). Cost modelling to support optimised selection of End-of-Life options for automotive components. *The International Journal of Advanced Manufacturing Technology, 73*(1-4), 399–407.

Zeithaml, V. A. (1988). Consumer Perceptions of Price, Quality, and Value: A Means-End Model and Synthesis of Evidence. *Journal of Marketing, 52*(3), 2–22.

Zell, A. (2003). *Simulation neuronaler Netze* (4., unveränd. Nachdr). München: Oldenbourg.

Zhou, J., Huang, P., Zhu, Y., & Deng, J. (2012). A quality evaluation model of reuse parts and its management system development for end-of-life wheel loaders. *Journal of Cleaner Production, 35*, 239–249.

Zink, T., & Geyer, R. (2017). Circular Economy Rebound. *Journal of Industrial Ecology, 21*(3), 593–602.

Zlamparet, G. I., Ijomah, W., Miao, Y., Awasthi, A. K., Zeng, X., & Li, J. (2017). Remanufacturing strategies: A solution for WEEE problem. *Journal of Cleaner Production, 149*, 126–136.

Appendix A

Source code data simulation in RStudio

Appendix A. 1: Import libraries and simulation of age and model

```
library(dplyr)
library(mosaic)
library(effects)
library(openxlsx)
library(GGally)
library(corrplot)
library(plyr)
library(pROC)
library(gplots)
library(ggplot)
library(sm)
library(fastDummies)

numberDatasets <- 250000

set.seed(1000)
age <- floor(rnorm(numberDatasets, mean = 36, sd = 8))
hist(age, main = "Histogram of age", xlab = "age in months", freq =
FALSE)

set.seed(2000)
model <- sample(x = c("Fujitsu Lifebook A357", "Fujitsu
                Lifebook U728", "Fujitsu Lifebook U729",
                "MacBook Pro", "MacBook Air", "HP EliteBook 830 G6",
                "HP EliteBook 850 G6","HP EliteBook 1040 G4",
                "HP EliteBook 1050 G1", "Dell Inspiron 13 5000",
                "Dell Precision 5520", "Lenovo Thinkpad T580",
                "Lenovo Thinkpad E490", "Acer Swift 1",
                "Acer Aspire 5", "Acer Aspire 7"),
                size = numberDatasets, replace = TRUE,
                prob =c(2/29, 1/29, 1/29, 4/29, 3/29, 2/29, 2/29,
                        1/29, 1/29, 2/29, 4/29, 1/29, 2/29, 1/29,
                        1/29, 1/29))
```

Table A- 1: Simulation of type of use I

```
set.seed(5100)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Fujitsu Lifebook A357")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.25, 0.30, 0.15, 0.05, 0.25))}}

set.seed(5200)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Fujitsu Lifebook U728")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive","gaming",
                     "videos"),
         size = 1, replace = TRUE,
         prob = c(0.08, 0.66, 0.18, 0.02, 0.16))}}
set.seed(5300)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Fujitsu Lifebook U729")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.08, 0.64, 0.20, 0.02, 0.16))}}

set.seed(5400)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "MacBook Pro") {df_model$typeOfUse[i]
  <-    sample(x = c("surfing", "normal business",
                     "computationally intensive”,
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.24, 0.40, 0.15, 0.01, 0.26))}}

set.seed(5500)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "MacBook Air")   {df_model$typeOfUse[i]
<-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.30, 0.32, 0.10, 0.01, 0.33))}}

set.seed(5600)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "HP EliteBook 830 G6")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.10, 0.58, 0.18, 0.04, 0.10))}}
```

Table A- 2: Simulation of type of use II

```
set.seed(5700)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "HP EliteBook 850 G6")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                               "gaming","videos"),
          size = 1, replace = TRUE,
          prob = c(0.18, 0.60, 0.18, 0.03, 0.11))}}
set.seed(5800)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "HP EliteBook 1040 G4")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive”,
                     "gaming","videos"),
          size = 1, replace = TRUE,
          prob = c(0.10, 0.68, 0.21, 0.05, 0.6))}}

for(i in 1:numberDatasets)
    {if(df_model$model[i] == "HP EliteBook 1050 G1")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
          size = 1, replace = TRUE,
          prob = c(0.03, 0.30, 0.50, 0.15, 0.02))}}

set.seed(6000)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Dell Inspiron 13 5000")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
          size = 1, replace = TRUE,
          prob = c(0.30, 0.30, 0.10, 0.02, 0.28))}}

set.seed(6100)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Dell Precision 5520")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
          size = 1, replace = TRUE,
          prob = c(0.07, 0.65, 0.24, 0.02, 0.12))}}

set.seed(6200)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Lenovo Thinkpad T580")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
          size = 1, replace = TRUE,
          prob = c(0.10, 0.63, 0.20, 0.02, 0.16))}}
```

Table A- 3: Simulation of type of use III

```
set.seed(6300)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Lenovo Thinkpad E490")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.29, 0.41, 0.16, 0.02, 0.24))}}

set.seed(6400)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Acer Swift 1")  {df_model$typeOfUse[i]
<-
        sample(x = c("surfing", "normal business",
                     "computationally intensive", gaming",
                     "videos"),
         size = 1, replace = TRUE,
         prob = c(0.46, 0.10, 0.06, 0.01, 0.42))}}

set.seed(6500)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Acer Aspire 5")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive",
                     "gaming","videos"),
         size = 1, replace = TRUE,
         prob = c(0.20, 0.40, 0.07, 0.03, 0.20))}}

set.seed(6600)
for(i in 1:numberDatasets)
    {if(df_model$model[i] == "Acer Aspire 7")
    {df_model$typeOfUse[i] <-
        sample(x = c("surfing", "normal business",
                     "computationally intensive", gaming",
                     "videos"),
         size = 1, replace = TRUE,
         prob = c(0.16, 0.42, 0.10, 0.03, 0.013))}}
```

Table A- 4: Simulation of mobile use

```
df_model <- mutate(df_model, mobileUse = 0)
set.seed(4000)
randomNumbers <- runif(numberDatasets, min = 0, max = 1)
df_model <- data.frame(df_model, randomNumbers = randomNumbers)
percentSurfingMobile <- 0.10
percentNormalBusinessMobile <- 0.40
percentComputationallyIntensiveMobile <- 0.18
percentGamingMobile <- 0.04
percentVideosMobile <- 0.12
df_model$mobileUse[df_model$typeOfUse ==
                   "surfing" & df_model$randomNumbers >
                   (1 - percentSurfingMobile)] = 1
df_model$mobileUse[df_model$typeOfUse ==
                   "normal business" & df_model$randomNumbers >
                   (1 - percentNormalBusinessMobile)] = 1
df_model$mobileUse[df_model$typeOfUse == "computationally intensive"
                   & df_model$randomNumbers >
                   (1 - percentComputationallyIntensiveMobile)] = 1
df_model$mobileUse[df_model$typeOfUse ==
                   "gaming" & df_model$randomNumbers >
                   (1 - percentGamingMobile)] = 1
df_model$mobileUse[df_model$typeOfUse ==
                   "videos" & df_model$randomNumbers >
                   (1 - percentVideosMobile)] = 1
df_model <- data.frame(df_model, mobileUse = df_model$mobileUse)
```

Table A- 5: Simulation of use of protective case

```
set.seed(5000)
randomNumbers <- runif(numberDatasets, min = 0, max = 1)
df_model <- mutate (df_model, useOfCover = 0)
df_model <- data.frame(df_model,randomNumbers = randomNumbers)
df_model <- data.frame(df_model, probCover = 0)
df_model$probCover = 0
df_model$probCover[df_model$typeOfUse == "surfing"] = 0.15
df_model$probCover[df_model$typeOfUse == "normal business"] = 0.35
df_model$probCover[df_model$typeOfUse == "computationally
                                          intensive"] = 0.25
df_model$probCover[df_model$typeOfUse == "gaming"] = 0.02
df_model$probCover[df_model$typeOfUse == "videos"] = 0.10

for(i in 1:numberDatasets) {if(df_model$mobileUse[i] == 1)
      {df_model$probCover[i] <- df_model$probCover[i] + 0.5}}

df_model$useOfCover[df_model$randomNumbers < df_model$probCover] = 1
```

Table A- 6: Simulation of price and brand

```
df_model <- data.frame(model = df_model$model,
                       typeOfUse = df_model$typeOfUse,
                       mobileUse = df_model$mobileUse,
                       useOfCover = df_model$useOfCover)

df_model1 <- data.frame(model = df_model$model,age = age,
                       typeOfUse =df_model$typeOfUse,
                       mobileUse = df_model$mobileUse,
                       useOfCover = df_model$useOfCover)

df_price <- data.frame(df_model1, price = 0)
df_price$price <- 0

df_price$price[df_price$model == "Fujitsu Lifebook A357"] = 479
df_price$price[df_price$model == "Fujitsu Lifebook U728"] = 1399
df_price$price[df_price$model == "Fujitsu Lifebook U729"] = 1299
df_price$price[df_price$model == "MacBook Pro"] = 1999
df_price$price[df_price$model == "MacBook Air"] = 1099
df_price$price[df_price$model == "HP EliteBook 830 G6"] = 1333
df_price$price[df_price$model == "HP EliteBook 850 G6"] = 1466
df_price$price[df_price$model == "HP EliteBook 1040 G4"] = 1796
df_price$price[df_price$model == "HP EliteBook 1050 G1"] = 3323
df_price$price[df_price$model == "Dell Inspiron 13 5000"] = 519
df_price$price[df_price$model == "Dell Precision 5520"] = 1906
df_price$price[df_price$model == "Lenovo Thinkpad T580"] = 1724
df_price$price[df_price$model == "Lenovo Thinkpad E490"] = 1011
df_price$price[df_price$model == "Acer Swift 1"] = 599
df_price$price[df_price$model == "Acer Aspire 5"] = 999
df_price$price[df_price$model == "Acer Aspire 7"] = 1299

df_brand <- data.frame(df_price, brand = "")
df_brand$brand <- ""

df_brand$brand[df_price$model == "Fujitsu Lifebook A357"] = "Fujitsu"
df_brand$brand[df_price$model == "Fujitsu Lifebook U728"] = "Fujitsu"
df_brand$brand[df_price$model == "Fujitsu Lifebook U729"] = "Fujitsu"
df_brand$brand[df_price$model == "MacBook Pro"] = "Apple"
df_brand$brand[df_price$model == "MacBook Air"] = "Apple"
df_brand$brand[df_price$model == "HP EliteBook 830 G6"] = "HP"
df_brand$brand[df_price$model == "HP EliteBook 850 G6"] = "HP"
df_brand$brand[df_price$model == "HP EliteBook 1040 G4"] = "HP"
df_brand$brand[df_price$model == "HP EliteBook 1050 G1"] = "HP"
df_brand$brand[df_price$model == "Dell Inspiron 13 5000"] = "Dell"
df_brand$brand[df_price$model == "Dell Precision 5520"] = "Dell"
df_brand$brand[df_price$model == "Lenovo Thinkpad T580"] = "Lenovo"
df_brand$brand[df_price$model == "Lenovo Thinkpad E490"] = "Lenovo"
df_brand$brand[df_price$model == "Acer Swift 1"] = "Acer"
df_brand$brand[df_price$model == "Acer Aspire 5"] = "Acer"
df_brand$brand[df_price$model == "Acer Aspire 7"] = "Acer"
```

Table A- 7: Simulation of external damages I

```
df_coverOutside <- data.frame(df_brand, coverOutside = "")
df_coverOutside$coverOutside <- ""

set.seed(7000)
for (i in 1:numberDatasets) {df_coverOutside$coverOutside[i] <-
        sample(x = c("All right", "Scratch", "Dent",
        "BrokenOutPart"),

                size = 1, replace = TRUE,
                prob = c(0.85, 0.08, 0.05, 0.02))}

df_display <- data.frame(df_coverOutside, display = "")
df_display$display <- ""

set.seed(7100)
for (i in 1:numberDatasets) {df_display$display[i] <-
        sample(x = c("All right", "Defect", "PixelErrors",
                     "Scratch", "BrokenScreen"),

                size = 1, replace = TRUE,
                prob = c(0.80, 0.05, 0.07, 0.03, 0.03))}

df_baseOutside <- data.frame(df_display, baseOutside = "" )
df_baseOutside$baseOutside <- ""

set.seed(7200)
for (i in 1:numberDatasets) {df_baseOutside$baseOutside[i] <-
        sample(x = c("All right", "Scratch", "Dent",
                     "BrokenOutPart"),

                size = 1, replace = TRUE,
                prob = c(0.85, 0.08, 0.05, 0.02))}

df_baseInside <- data.frame(df_baseOutside, baseInside = "" )
df_baseInside$baseInside <- ""

set.seed(7300)
for (i in 1:numberDatasets) {df_baseInside$baseInside[i] <-
        sample(x = c("All right", "Scratch", "Dent",
                     "BrokenOutPart"),

                size = 1, replace = TRUE,
                prob = c(0.87, 0.10, 0.02, 0.01))}

df_keyboard <- data.frame(df_baseInside, keyboard = "" )
df_keyboard$keyboard <- ""

set.seed(7400)
for (i in 1:numberDatasets) {df_keyboard$keyboard[i] <-
        sample(x = c("All right", "Highly contaminated",
                     "Missing key", "Defect"),

                size = 1, replace = TRUE,
                prob = c(0.79, 0.15, 0.02, 0.04))}

```

Table A- 8: Simulation of external damages II

```
df_mousepad <- data.frame(df_keyboard, mousepad = "" )
df_mousepad$mousepad <- ""

set.seed(7500)
for (i in 1:numberDatasets) {df_mousepad$mousepad[i] <-
        sample(x = c("All right", "Scratch", "Defect"),

                    size = 1, replace = TRUE,
                    prob = c(0.85, 0.12, 0.03))}

```

Table A- 9: Creating dummy variables

```
vectorCategoricalVariables <- c("model", "typeOfUse", "brand",
                                "coverOutside", "display",
                                "baseOutside", "baseInside",
                                "keyboard", "mousepad")

datadummies <- dummy_cols(df_mousepad, select_columns =
                          vectorCategoricalVariables)

```

Table A- 10: Defining the betas

```
Intercept <- -9.2; betaAge <- 0.18; betaMobileUse <- 3;
betaUseOfCover <- -1.0; betaPrice <- -0.0011;

betaModelLifebookA357 <- 0; betaModelLifebookU728 <- 0 ;
betaModelLifebookU729 <- 0;
betaModelMacBookPro <- 0; betaModelMacBookAir <- 0;
betaModelHPEliteBook830G6 <- 0;
betaModelHPEliteBook850G6 <- 0; betaModelHPEliteBook1040G4 <- 0;
betaModelHPEliteBook1050G1 <- 0
betaModelDellInspirion135000 <- 0; betaModelDellPrecision5520 <- 0;
betaModelLenovoThinkpadT580 <- 0;
betaModelLenovoThinkpadE490 <- 0; betaModelAcerSwift1 <- 0;
betaModelAcerAspire5 <- 0; betaModelAcerAspire7 <- 0

betaTypeOfUseSurfing <- 0; betaTypeOfUseBusiness <- 0;
betaTypeOfUseIntensive <- 6;
betaTypeOfUseGaming <- 5; betaTypeOfUseVideos <- 0

betaBrandFujitsu <- 0; betaBrandApple <- 0; betaBrandHP <- 0;
betaBrandDell <- 0;
betaBrandLenovo <- 0; betaBrandAcer <- 0

betaCoverOutsideAllright <- 0; betaCoverOutsideScratch <- 0;
betaCoverOutsideDent <-0; betaCoverOutsideBroken <- 1.1

betaDisplayAllright <- 0; betaDisplayDefect <- 0;betaDisplayScratch
<-0;
betaDisplayBrokenScreen <- 1.2; betaDisplayPixelErrors <- 0

betaBaseOutsideAllright <- 0; betaBaseOutsideScratch <- 0;
betaBaseOutsideDent <- 0.7; betaBaseOutsideBrokenOut <- 1.5

betaBaseInsideallright <- 0; betaBaseInsideScratch <- 0;
betaBaseInsideDent <- 0.7; betaBaseInsideBrokenOut <- 1.5

betaKeyboardAllright <- 0; betaKeyboardMissing <- 0;
betaKeyboardHighlyContaminated <- 0; betaKeyboardDefect <- 0

betaMousePadAllright <- 0; betaMousePadScratch <- 0;
betaMousePadDefect <- 0.4
```

Table A- 11: Calculating the dependent variables

```
df_prob1 <-
        mutate(datadummies, probBattery = 1/(1+exp(-(Intercept +
                (betaAge * age) + (betaMobileUse * mobileUse)
                + (betaUseOfCover * useOfCover) + (betaPrice *
                    price)
                + (betaModelLifebookA357 * `model_Fujitsu Lifebook
                    A357`)
                + (betaModelLifebookU728 *
                datadummies$`model_Fujitsu Lifebook U728`)
                + (betaModelLifebookU729 *
                datadummies$`model_Fujitsu Lifebook U729`)
                + (betaModelMacBookPro * datadummies$`model_MacBook
                    Pro`)
                + (betaModelMacBookAir * datadummies$`model_MacBook
                    Air`)
                + (betaModelHPEliteBook830G6 * datadummies$`model_HP
                    EliteBook 830 G6`)
                + (betaModelHPEliteBook850G6 * datadummies$`model_HP
                    EliteBook 850 G6`)
                + (betaModelHPEliteBook1040G4 *
                datadummies$`model_HP EliteBook 1040 G4`)
                + (betaModelHPEliteBook1050G1 *
                datadummies$`model_HP EliteBook 1050 G1`)
                + (betaModelDellInspirion135000 *
                    datadummies$`model_Dell Inspiron 13 5000`)
                + (betaModelDellPrecision5520 *
                    datadummies$`model_Dell Precision 5520`)
                + (betaModelLenovoThinkpadT580 *
                    datadummies$`model_Lenovo Thinkpad T580`)
                + (betaModelLenovoThinkpadE490 *
                    datadummies$`model_Lenovo Thinkpad E490`)
                + (betaModelAcerSwift1 * datadummies$`model_Acer
                    Swift 1`)
                + (betaModelAcerAspire5 * datadummies$`model_Acer
                    Aspire 5`)
                + (betaModelAcerAspire7 * datadummies$`model_Acer
                    Aspire 7`)
                + (betaTypeOfUseSurfing *
                    datadummies$typeOfUse_surfing)
                + (betaTypeOfUseBusiness *
                    datadummies$`typeOfUse_normal business`)
                + (betaTypeOfUseIntensive * datadummies$`
                    typeOfUse_computationally intensive`)
                + (betaTypeOfUseGaming
                    * datadummies$typeOfUse_gaming)
                + (betaTypeOfUseVideos
                    * datadummies$typeOfUse_gaming)
                + (betaCoverOutsideAllright
                    * datadummies$`coverOutside_All right`)
                + (betaCoverOutsideScratch
                    * datadummies$coverOutside_Scratch)
                + (betaCoverOutsideDent
                    * datadummies$coverOutside_Dent)
                + (betaCoverOutsideBroken
                    * datadummies$coverOutside_BrokenOutPart)
                + (betaDisplayAllright
                    * datadummies$`display_All right`)
                + (betaDisplayDefect * datadummies$display_Defect)
                + (betaDisplayPixelErrors
```

```
                         * datadummies$display_PixelErrors)
                    + (betaDisplayScratch * datadummies$display_Scratch)
                    + (betaDisplayBrokenScreen
                         * datadummies$display_BrokenScreen)
                    + (betaBaseOutsideAllright
                         * datadummies$`baseOutside_All right`)
                    + (betaBaseOutsideScratch
                         * datadummies$baseOutside_Scratch)
                    + (betaBaseOutsideDent
                         * datadummies$baseOutside_Dent)
                    + (betaBaseOutsideBrokenOut
                         * datadummies$baseOutside_BrokenOutPart)
                    + (betaBaseInsideallright
                         * datadummies$`baseInside_All right`)
                    + (betaBaseInsideScratch
                         * datadummies$baseInside_Scratch)
                    + (betaBaseInsideDent * datadummies$baseInside_Dent)
                    + (betaBaseInsideBrokenOut
                         * datadummies$baseInside_BrokenOutPart)
                    + (betaKeyboardAllright
                         * datadummies$`keyboard_All right`)
                    + (betaKeyboardMissing
                         * datadummies$`keyboard_Missing key`)
                    + (betaKeyboardHighlyContaminated
                         * datadummies$`keyboard_Highly contaminated`)
                    + (betaKeyboardDefect * datadummies$keyboard_Defect)
                    + (betaMousePadAllright
                         * datadummies$`mousepad_All right`)
                    + (betaMousePadScratch
                         * datadummies$mousepad_Scratch)
                    + (betaMousePadDefect * datadummies$mousepad_Defect)
                ))))

set.seed(1050)
randomNumbers <- runif(numberDatasets, min = 0, max = 1)
df_prob1 <- mutate(df_prob1, conditionBattery = 0, randomNumbers =
                       randomNumbers)
df_prob1$conditionBattery <- 0
df_prob1$conditionBattery[df_prob1$probBattery >=
                            df_prob1$randomNumbers] = 1

vars <- colnames(df_prob1)
varDelete <- c("randomNumbers", "probBattery")
df_prob1 <- df_prob1[, !(vars %in% varDelete)]
```

This shows the calculation of the condition of the battery. This is done in the same way for the nine other components, but with different values for the beta-factors.

Appendix B

Source code executive program in Python

Table B- 1: Import of required libraries and definition of global variables

```
import urllib.request
import json
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton,
                             QToolTip, QLabel, QRadioButton,
                             QComboBox, QLineEdit, QSpinBox
from PyQt5.QtGui import QIcon, QFont, QIntValidator
from PyQt5 import QtCore
from PyQt5.QtCore import Qt, QObject, pyqtSignal

profitMargin = 0.20
riskProportion = 0.10
interestRate = 0.05
averageStorageTime = 0.25
hourlyRateRepair = 50
labourHoursRefurbishing = 0.75
hourlyRateRefurbishing = 35
overHeadRateManufacturing = 0.1
overHeadRateMaterial = 0.1
totalEmissionsLaptop = 214
emissionsSpareBattery = totalEmissionsLaptop * 0.07
emissionsSpareCPU = totalEmissionsLaptop * 0.16
emissionsSpareGPU = totalEmissionsLaptop * 0.05
emissionsSpareDisplay = totalEmissionsLaptop * 0.25
emissionsSpareKeyboard = totalEmissionsLaptop * 0.02
emissionsSpareMousepad = totalEmissionsLaptop * 0.02
emissionsSpareMotherboard = totalEmissionsLaptop * 0.20
emissionsSpareOuterCasing = totalEmissionsLaptop * 0.16
emissionsSpareHarddrive = totalEmissionsLaptop * 0.02
emissionsSpareCoolingSystem = totalEmissionsLaptop * 0.02
```

Table B- 2: Definition of class 'LaptopModel'

```
class LaptopModel(object):
    def __init__(self, id, model, brand, price, sellingPrice, cpu,
                 gpu, coolingSystem, battery, display,
                 keyboard, mousepad, motherboard, outerCasing,
                 harddrive):
        self.ID = id
        self.Model = model
        self.Brand = brand
        self.NewPrice = price
        self.SellingPrice = sellingPrice
        self.CPU = cpu
        self.GPU = gpu
        self.CoolingSystem = coolingSystem
        self.Battery = battery
        self.Display = display
        self.Keyboard = keyboard
        self.Mousepad = mousepad
        self.Motherboard = motherboard
        self.OuterCasing = outerCasing
        self.Harddrive = harddrive
```

Table B- 3: Definition of class 'PotentiaLaptop'

```
class PotentialLaptop(object):
    def __init__(self, model, age, typeOfUse, mobileUse, useOfCase,
                 newPrice, brand, topCasingOutside, display,
                 bottomCasingOutside,bottomCasingInside,
                 keyboard, mousepad):
        self.Model = model
        self.Age = age
        self.TypeOfUse = typeOfUse
        self.MobileUse = mobileUse
        self.UseOfCase = useOfCase
        self.NewPrice = newPrice
        self.Brand = brand
        self.TopCasingOutside = topCasingOutside
        self.Display = display
        self.BottomCasingOutside = bottomCasingOutside
        self.BottomCasingInside = bottomCasingInside
        self.Keyboard = keyboard
        self.Mousepad = mousepad
```

Table B- 4: Definition class 'UsedLaptop'

```
class UsedLaptop(object):
    def __init__(self, id, model, brand, location, cpuCond, gpuCond,
                 coolingSystemCond,batteryCond, displayCond,
                 keyboardCond, mousepadCond, motherboardCond,
                 outerCasingCond, harddriveCond):
        self.ID = id
        self.Model = model
        self.Brand = brand
        self.Location = location
        self.CPUCond = cpuCond
        self.GPUCond = gpuCond
        self.CoolingSystemCond = coolingSystemCond
        self.BatteryCond = batteryCond
        self.DisplayCond = displayCond
        self.KeyboardCond = keyboardCond
        self.MousepadCond = mousepadCond
        self.MotherboardCond = motherboardCond
        self.OuterCasingCond = outerCasingCond
        self.HarddriveCond = harddriveCond
```

Table B- 5: Definition class 'repair'

```
class repair(object):
    def __init__(self, rid, model, component, sparepartprice,
                 labourhours, repairCost):
        self.ID = rid
        self.Model = model
        self.Component = component
        self.SparepartPrice = sparepartprice
        self.LabourHours = labourhours
        self.PriceLabour = hourlyRateRepair * labourhours
        if (repairCost != 0):
            self.RepairCost = repairCost * 0.8
        else:
            self.RepairCost = self.PriceLabour + sparepartprice
```

Table B- 6: Creating laptop model objects I

```
lifebookA357 = LaptopModel(0, "Fujitsu Lifebook A357", "Fujitsu",
                          479, 299, "8 GB DDR4 2133 MHz SODIMM",
                          "Intel HD Graphics 620",
                          "CoolingSystem Lifebook A357",
                          "Battery Lifebook A357",
                          "Display Lifebook A357",
                          "Keyboard Lifebook A357",
                          "Mousepad Lifebook A357",
                          "Motherboard Lifebook A357",
                          "Casing Lifebook A357",
                          "256 GB SSD SATA III")

lifebookU728 = LaptopModel(1, "Fujitsu Lifebook U728", "Fujitsu",
                          1399, 840, "8 GB DDR4 2133 MHz SODIMM",
                          "GPU Lifebook U728",
                          "CoolingSystem Lifebook U728",
                          "Battery Lifebook U728",
                          "Display Lifebook U728",
                          "Keyboard Lifebook U728",
                          "Mousepad Lifebook U728",
                          "Motherboard Lifebook U728",
                          "Case Lifebook U728",
                          "256 GB SSD SATA III")

lifebookU729 = LaptopModel(2, "Fujitsu Lifebook U729", "Fujitsu",
                          1299, 780, "8 GB DDR4 2133 MHz SODIMM",
                          "GPU Lifebook U729",
                          "CoolingSystem Lifebook U729",
                          "Battery Lifebook U729",
                          "Display Lifebook U729",
                          "Keyboard Lifebook U729",
                          "Mousepad Lifebook U729",
                          "Motherboard Lifebook U729",
                          "Case Lifebook U729", "256 GB PCIe SSD")

macBookPro = LaptopModel(3, "MacBook Pro", "Apple", 1999, 1399,
                         "8 GB LPDDR3 2133 MHz", "GPU MacBook Pro",
                         "CoolingSystem MacBook Pro",
                         "Battery MacBook Pro",
                         "Display MacBook Pro",
                         "Keyboard MacBook Pro",
                         "Mousepad MacBook Pro",
                         "Motherboard MacBook Pro",
                         "Case MacBook Pro", "256 GB SSD")

macBookAir = LaptopModel(4, "MacBook Air", "Apple", 1099, 769,
                         "8 GB LPDDR3 1600 MHz", "GPU MacBook Air",
                         "CoolingSystem MacBook Air",
                         "Battery MacBook Air",
                         "Display MacBook Air",
                         "Keyboard MacBook Air",
                         "Mousepad MacBook Air",
                         "Motherboard MacBook Air",
                         "Case MacBook Air", "128 GB SSD")
```

Table B- 7: Creating laptop model objects II

```
hpEliteBook830G6 = LaptopModel(5, "HP EliteBook 830 G6", "HP", 1333,
                               799, "8 GB SDRAM", "GPU EliteBook830",
                               "CoolingSystem EliteBook 830 G6",
                               "Battery EliteBook 830 G6",
                               "Display EliteBook 830 G6",
                               "Keyboard EliteBook 830 G6",
                               "Mousepad EliteBook 830 G6",
                               "Motherboard EliteBook 830 G6",
                               "Case EliteBook 830 G6",
                               "256 GB PCIe SSD")

hpEliteBook850G6 = LaptopModel(6, "HP EliteBook 850 G6", "HP", 1466,
                               879, "8 GB SDRAM", "GPU EliteBook850",
                               "CoolingSystem EliteBook 850 G6",
                               "Battery EliteBook 850 G6",
                               "Display EliteBook 850 G6",
                               "Keyboard EliteBook 850 G6",
                               "Mousepad EliteBook 850 G6",
                               "Motherboard EliteBook 850 G6",
                               "Case EliteBook 850 G6",
                               "256 GB PCIe SSD")

hpEliteBook1040G4 = LaptopModel(7, "HP EliteBook 1040 G4", "HP",
                                1796, 1077, "8 GB SDRAM",
                                "GPU EliteBook1040",
                                "CoolingSystem EliteBook 1040 G4",
                                "Battery EliteBook 1040 G4",
                                "Display EliteBook 1040 G4",
                                "Keyboard EliteBook 1040 G4",
                                "Mousepad EliteBook 1040 G4",
                                "Motherboard EliteBook 1040 G4",
                                "Case EliteBook 1040 G4",
                                "256 HP Z Turbo PCIe SSD")

hpEliteBook1050G1 = LaptopModel(8, "HP EliteBook 1050 G1", "HP",
                                3323, 1999, "32 GB SDRAM",
                                "GPU EliteBook1050",
                                "CoolingSystem EliteBook 1050 G1",
                                "Battery EliteBook 1050 G1",
                                "Display EliteBook 1050 G1",
                                "Keyboard EliteBook 1050 G1",
                                "Mousepad EliteBook 1050 G1",
                                "Motherboard EliteBook 1050 G1",
                                "Case EliteBook 1050 G1",
                                "1 TB PCIe SSD")

inspirion135000 = LaptopModel(9, "Dell Inspirion 13 5000", "Dell",
                              519, 311, "4 GB DDR4 2400 MHz",
                              "GPU Inspition135000",
                              "CoolingSystem Inspirion 13 5000",
                              "Battery Inspirion 13 5000",
                              "Display Inspirion 13 5000",
                              "Keyboard Inspirion 13 5000",
                              "Mousepad Inspirion 13 5000",
                              "Motherboard Inspirion 13 5000",
                              "Case Inspirion 13 5000",
                              "128 GB SSD")
```

Table B- 8: Creating laptop model objects III

```
precision5520 = LaptopModel(10, "Dell Precision 5520", "Dell", 1906,
                            1149, "8 GB DDR4 SDRAM",
                            "GPU Precision5520",
                            "CoolingSystem Precision 5520",
                            "Battery Precision 5520",
                            "Display Precision 5520",
                            "Keyboard Precision 5520",
                            "Mousepad Precision 5520",
                            "Motherboard Precision 5520",
                            "Case Precision 5520", "256 GB SSD")

thinkpadT580 = LaptopModel(11, "Dell Thinkpad T580", "Dell", 1724,
                           1034, "16 GB DDR4 2133 MHz SODIMM",
                           "GPU ThinkpadT580",
                           "CoolingSystem Thinkpad T580",
                           "Battery Thinkpad T580",
                           "Display Thinkpad T580",
                           "Keyboard Thinkpad T580",
                           "Mousepad Thinkpad T580",
                           "Motherboard Thinkpad T580",
                           "Case Thinkpad T580", "512 GB PCIe SSD")

thinkpadE490 = LaptopModel(12, "Dell Thinkpad E490", "Dell", 1011,
                           606, "16 GB DDR4 2133 MHz SODIMM",
                           "GPU ThinkpadE490",
                           "CoolingSystem Thinkpad T580",
                           "Battery Thinkpad T580",
                           "Display Thinkpad T580",
                           "Keyboard Thinkpad T580",
                           "Mousepad Thinkpad T580",
                           "Motherboard Thinkpad T580",
                           "Case Thinkpad T580", "512 GB PCIe SSD")

acerSwift1 = LaptopModel(13, "Acer Swift 1", "Acer", 599, 359,
                         "4 GB SDRAM", "GPU Swift 1",
                         "CoolingSystem Swift 1", "Battery Swift 1",
                         "Display Swift 1","Keyboard Swift 1",
                         "Mousepad Swift 1", "Motherboard Swift 1",
                         "Case Swift 1", "256 GB SSD")

acerAspire5 = LaptopModel(14, "Acer Aspire 5", "Acer", 999, 599,
                          "8 GB SDRAM", "GPU Aspire 5",
                          "CoolingSystem Aspire 5",
                          "Battery Aspire 5", "Display Aspire 5",
                          "Keyboard Aspire 5", "Mousepad Aspire 5",
                          "Motherboard Aspire 5",
                          "Case Aspire 5", "256 GB SSD")

acerAspire7 = LaptopModel(15, "Acer Aspire 7", "Acer", 1299, 779,
                          "16 GB SDRAM", "GPU Aspire 7",
                          "CoolingSystem Aspire 7",
                          "Battery Aspire 7", "Display Aspire 7",
                          "Keyboard Aspire 7", "Mousepad Aspire 7",
                          "Motherboard Aspire 7","Case Aspire 7",
                          "256 GB SSD")
```

Table B- 9: Creating list of laptop models

```
listLaptopModels = [lifebookA357, lifebookU728, lifebookU729,
                    macBookPro, macBookAir, hpEliteBook830G6,
                    hpEliteBook850G6, hpEliteBook1040G4,
                    hpEliteBook1050G1, inspirion135000,
                    precision5520,thinkpadE490, thinkpadT580,
                    acerSwift1, acerAspire5, acerAspire7]
```

Table B- 10: Creating objects of 'repair' I

```
repairLifebookA357CPU = repair(0, "Lifebook A357", "CPU",
                              56, 0.7,0)
repairLifebookA357Battery = repair(1, "Lifebook A357", "Battery",
                                   89, 0.5, 0)
repairLifebookA357Harddrive = repair(2, "Lifebook A357",
                                     "Harddrive", 56, 0.7, 0)
repairLifebookA357GPU = repair(3, "Lifebook A357", "GPU", 0, 1, 399)
repairLifebookA357CoolingSystem = repair(4, "Lifebook A357",
                                     "CoolingSystem", 48, 0.7, 0)
repairLifebookA357Keyboard = repair(5, "Lifebook A357", "Keyboard",
                                     52, 0.8, 0)
repairLifebookA357Motherboard = repair(6, "Lifebook A357",
                                     "Motherboard", 0, 1, 399)
repairLifebookA357Mousepad = repair(7, "Lifebook A357", "Mousepad",
                                     35, 0.8, 0)
repairLifebookA357Display = repair(8, "Lifebook A357", "Display",
                                     181, 1.2, 0)
repairLifebookU728CPU = repair(10, "Lifebook U728", "CPU",
                              56, 0.7, 0)
repairLifebookU728Battery = repair(11, "Lifebook U728", "Battery",
                                      125, 0.5, 0)
repairLifebookU728Harddrive = repair(12, "Lifebook U728",
                                      "Harddrive", 56, 0.7, 0)
repairLifebookU728GPU = repair(13, "Lifebook U728", "GPU",
                              0, 1, 479)
repairLifebookU728CoolingSystem = repair(14, "Lifebook U728",
                                      "CoolingSystem", 48, 0.7, 0)
repairLifebookU728Keyboard = repair(15, "Lifebook U728", "Keyboard",
                                      106, 0.8, 0)
repairLifebookU728Motherboard = repair(16, "Lifebook U728",
                                      "Motherboard", 0, 1, 479)
repairLifebookU728Mousepad = repair(17, "Lifebook U728", "Mousepad",
                                      42, 0.8, 0)
repairLifebookU728Display = repair(18, "Lifebook U728", "Display",
                                      268, 1.2, 0)
```

Table B- 11: Creating objects of 'repair' II

```
repairLifebookU729CPU = repair(20, "Lifebook U729", "CPU",
                               56, 0.7, 0)
repairLifebookU729Battery = repair(21, "Lifebook U729", "Battery",
                               125, 0.5, 0)
repairLifebookU729Harddrive = repair(22, "Lifebook U729",
                               "Harddrive", 69, 0.7, 0)
repairLifebookU729GPU = repair(23, "Lifebook U729", "GPU",
                               0, 1, 479)
repairLifebookU729CoolingSystem = repair(24, "Lifebook U729",
                               "CoolingSystem", 48, 0.7, 0)
repairLifebookU729Keyboard = repair(25, "Lifebook U729", "Keyboard",
                               106, 0.8, 0)
repairLifebookU729Motherboard = repair(26, "Lifebook U729",
                               "Motherboard", 0, 1, 479)
repairLifebookU729Mousepad = repair(27, "Lifebook U729", "Mousepad",
                               42, 0.8, 0)
repairLifebookU729Display = repair(28, "Lifebook U729", "Display",
                               287, 1.2, 0)
repairMacBookProCPU = repair(30, "MacBook Pro", "CPU", 66, 0.8, 0)
repairMacBookProBattery = repair(31, "MacBook Pro", "Battery",
                               0, 0.6, 249)
repairMacBookProHarddrive = repair(32, "MacBook Pro", "Harddrive",
                               76, 0.8, 0)
repairMacBookProGPU = repair(33, "MacBook Pro", "GPU", 0, 1, 499)
repairMacBookProCoolingSystem = repair(34, "MacBook Pro",
                               "CoolingSystem", 58, 0.8, 0)
repairMacBookProKeyboard = repair(35, "MacBook Pro", "Keyboard",
                               0, 1, 229)
repairMacBookProMotherboard = repair(36, "MacBook Pro",
                               "Motherboard", 159, 3.2, 0)
repairMacBookProMousepad = repair(37, "MacBook Pro", "Mousepad",
                               0, 0, 299)
repairMacBookProDisplay = repair(38, "MacBook Pro", "Display",
                               0, 0, 480)
repairMacBookAirCPU = repair(40, "MacBook Air", "CPU", 56, 0.8, 0)
repairMacBookAirBattery = repair(41, "MacBook Air", "Battery",
                               0, 1, 159)
repairMacBookAirHarddrive = repair(42, "MacBook Air", "Harddrive",
                               56, 0.8, 0)
repairMacBookAirGPU = repair(43, "MacBook Air", "GPU", 0, 1, 399)
repairMacBookAirCoolingSystem = repair(44, "MacBook Air",
                               "CoolingSystem", 65, 0.8, 0)
repairMacBookAirKeyboard = repair(45, "MacBook Air", "Keyboard",
                               0, 1, 119)
repairMacBookAirMotherboard = repair(46, "MacBook Air",
                               "Motherboard", 139, 3.2, 0)
repairMacBookAirMousepad = repair(47, "MacBook Air", "Mousepad",
                               0, 1, 149)
repairMacBookAirDisplay = repair(48, "MacBook Air", "Display",
                               0, 1, 379)

```

Table B- 12: Creating objects of 'repair' III

```
repairhpEliteBook830G6CPU = repair(50, "EliteBook 830 G6", "CPU",
                                   56, 0.7, 0)
repairhpEliteBook830G6Battery = repair(51, "EliteBook 830 G6",
                                   "Battery", 75, 0.5, 0)
repairhpEliteBook830G6Harddrive = repair(52, "EliteBook 830 G6",
                                   "Harddrive", 69, 0.7, 0)
repairhpEliteBook830G6GPU = repair(53, "EliteBook 830 G6", "GPU",
                                   0, 1, 399)
repairhpEliteBook830G6CoolingSystem = repair(54, "EliteBook 830 G6",
                                   "CoolingSystem", 99, 0.7, 0)
repairhpEliteBook830G6Keyboard = repair(55, "EliteBook 830 G6",
                                   "Keyboard", 114, 0.8, 0)
repairhpEliteBook830G6Motherboard = repair(56, "EliteBook 830 G6",
                                   "Motherboard", 0, 1, 479)
repairhpEliteBook830G6Mousepad = repair(57, "EliteBook 830 G6",
                                   "Mousepad", 35, 0.8, 0)
repairhpEliteBook830G6Display = repair(58, "EliteBook 830 G6",
                                   "Display", 0, 1, 199)
repairhpEliteBook850G6CPU = repair(60, "EliteBook 850 G6", "CPU",
                                   56, 0.7, 0)
repairhpEliteBook850G6Battery = repair(61, "EliteBook 850 G6",
                                   "Battery", 0.75, 0.5, 0)
repairhpEliteBook850G6Harddrive = repair(62, "EliteBook 850 G6",
                                   "Harddrive", 69, 0.7, 0)
repairhpEliteBook850G6GPU = repair(63, "EliteBook 850 G6", "GPU",
                                   0, 1, 399)
repairhpEliteBook850G6CoolingSystem = repair(64, "EliteBook 850 G6",
                                   "CoolingSystem", 99, 0.7, 0)
repairhpEliteBook850G6Keyboard = repair(65, "EliteBook 850 G6",
                                   "Keyboard", 114, 0.8, 0)
repairhpEliteBook850G6Motherboard = repair(66, "EliteBook 850 G6",
                                   "Motherboard", 0, 1, 479)
repairhpEliteBook850G6Mousepad = repair(67, "EliteBook 850 G6",
                                   "Mousepad", 55, 0.8, 0)
repairhpEliteBook850G6Display = repair(68, "EliteBook 850 G6",
                                   "Display", 0, 1, 199)
repairhpEliteBook1040G4CPU = repair(70, "EliteBook 1040 G4", "CPU",
                                   56, 0.7, 0)
repairhpEliteBook1040G4Battery = repair(71, "EliteBook 1040 G4",
                                   "Battery", 99, 0.5, 0)
repairhpEliteBook1040G4Harddrive = repair(72, "EliteBook 1040 G4",
                                   "Harddrive", 160, 0.7, 0)
repairhpEliteBook1040G4GPU = repair(73, "EliteBook 1040 G4", "GPU",
                                   0, 1, 479)
repairhpEliteBook1040G4CoolingSystem = repair(74,
                                   "EliteBook 1040 G4",
                                   "CoolingSystem", 89, 0.7, 0)
repairhpEliteBook1040G4Keyboard = repair(75, "EliteBook 1040 G4",
                                   "Keyboard", 114, 0.8, 0)
repairhpEliteBook1040G4Motherboard = repair(76, "EliteBook 1040 G4",
                                   "Motherboard", 0, 1, 479)
repairhpEliteBook1040G4Mousepad = repair(77, "EliteBook 1040 G4",
                                   "Mousepad", 55, 0.8, 0)
repairhpEliteBook1040G4Display = repair(78, "EliteBook 1040 G4",
                                   "Display", 0, 1, 299)
```

Table B- 13: Creating objects of 'repair' IV

```
repairhpEliteBook1050G1CPU = repair(80, "EliteBook 1050 G1", "CPU",
                                       56, 0.7, 0)
repairhpEliteBook1050G1Battery = repair(81, "EliteBook 1050 G1",
                                       "Battery", 99, 0.5, 0)
repairhpEliteBook1050G1Harddrive = repair(82, "EliteBook 1050 G1",
                                       "Harddrive", 214, 0.7, 0)
repairhpEliteBook1050G1GPU = repair(83, "EliteBook 1050 G1", "GPU",
                                       0, 1, 479)
repairhpEliteBook1050G1CoolingSystem = repair(84, "EliteBook 1050
G1",                                   "CoolingSystem", 89, 0.7, 0)
repairhpEliteBook1050G1Keyboard = repair(85, "EliteBook 1050 G1",
                                       "Keyboard", 114, 0.8, 0)
repairhpEliteBook1050G1Motherboard = repair(86, "EliteBook 1050 G1",
                                       "Motherboard", 0, 1, 399)
repairhpEliteBook1050G1Mousepad = repair(87, "EliteBook 1050 G1",
                                       "Mousepad", 55, 0.8, 0)
repairhpEliteBook1050G1Display = repair(88, "EliteBook 1050 G1",
                                       "Display", 0, 1, 299)
repairInspirion135000CPU = repair(90, "Inspirion 13 5000", "CPU",
                                       46, 0.7, 0)
repairInspirion135000Battery = repair(91, "Inspirion 13 5000",
                                       "Battery", 69, 0.5, 0)
repairInspirion135000Harddrive = repair(92, "Inspirion 13 5000",
                                       "Harddrive", 45, 0.7, 0)
repairInspirion135000GPU = repair(93, "Inspirion 13 5000", "GPU",
                                       0, 1, 299)
repairInspirion135000CoolingSystem = repair(94, "Inspirion 13 5000",
                                       "CoolingSystem", 48, 0.7, 0)
repairInspirion135000Keyboard = repair(95, "Inspirion 13 5000",
                                       "Keyboard", 58, 0.8, 0)
repairInspirion135000Motherboard = repair(96, "Inspirion 13 5000",
                                       "Motherboard", 0, 1, 299)
repairInspirion135000Mousepad = repair(97, "Inspirion 13 5000",
                                       "Mousepad", 35, 0.8, 0)
repairInspirion135000Display = repair(98, "Inspirion 13 5000",
                                       "Display", 0, 1, 0)
repairPrecision5520CPU = repair(100, "Precision 5520", "CPU",
                                       56, 0.7, 0)
repairPrecision5520Battery = repair(101, "Precision 5520",
                                       "Battery", 79, 0.5, 0)
repairPrecision5520Harddrive = repair(102, "Precision 5520",
                                       "Harddrive", 59, 0.7, 0)
repairPrecision5520GPU = repair(103, "Precision 5520", "GPU",
                                       0, 1, 479)
repairPrecision5520CoolingSystem = repair(104, "Precision 5520",
                                       "CoolingSystem", 48, 0.7, 0)
repairPrecision5520Keyboard = repair(105, "Precision 5520",
                                       "Keyboard", 114, 0.8, 0)
repairPrecision5520Motherboard = repair(106, "Precision 5520",
                                       "Motherboard", 0, 1, 399)
repairPrecision5520Mousepad = repair(107, "Precision 5520",
                                       "Mousepad", 45, 0.8, 0)
repairPrecision5520Display = repair(108, "Precision 5520",
                                       "Display", 0, 1, 279)
```

Table B- 14: Creating objects of 'repair' V

```
repairThinkpadT580CPU = repair(110, "Thinkpad T580", "CPU",
                                       89, 0.7, 0)
repairThinkpadT580Battery = repair(111, "Thinkpad T580", "Battery",
                                       87, 0.5, 0)
repairThinkpadT580Harddrive = repair(112, "Thinkpad T580",
                                       "Harddrive", 104, 0.7, 0)
repairThinkpadT580GPU = repair(113, "Thinkpad T580", "GPU",
                                       0, 1, 479)
repairThinkpadT580CoolingSystem = repair(114, "Thinkpad T580",
                                       "CoolingSystem", 59, 0.7, 0)
repairThinkpadT580Keyboard = repair(115, "Thinkpad T580",
                                       "Keyboard", 65, 0.8, 0)
repairThinkpadT580Motherboard = repair(116, "Thinkpad T580",
                                       "Motherboard", 0, 1, 479)
repairThinkpadT580Mousepad = repair(117, "Thinkpad T580",
                                       "Mousepad", 45, 0.8, 0)
repairThinkpadT580Display = repair(118, "Thinkpad T580", "Display",
                                       167, 1.2, 0)
repairThinkpadE490CPU = repair(120, "Thinkpad E490", "CPU",
                                       89, 0.7, 0)
repairThinkpadE490Battery = repair(121, "Thinkpad E490", "Battery",
                                       99, 0.5, 0)
repairThinkpadE490Harddrive = repair(122, "Thinkpad E490",
                                       "Harddrive", 104, 0.7, 0)
repairThinkpadE490GPU = repair(123, "Thinkpad E490", "GPU",
                                       0, 1, 399)
repairThinkpadE490CoolingSystem = repair(124, "Thinkpad E490",
                                       "CoolingSystem", 59, 0.7, 0)
repairThinkpadE490Keyboard = repair(125, "Thinkpad E490",
                                       "Keyboard", 73, 0.8, 0)
repairThinkpadE490Motherboard = repair(126, "Thinkpad E490",
                                       "Motherboard", 0, 1, 399)
repairThinkpadE490Mousepad = repair(127, "Thinkpad E490",
                                       "Mousepad", 35, 0.8, 0)
repairThinkpadE490Display = repair(128, "Thinkpad E490", "Display",
                                       0, 1, 199)
repairSwift1CPU = repair(130, "Swift 1", "CPU", 39, 0.7, 0)
repairSwift1Battery = repair(131, "Swift 1", "Battery", 82, 0.5, 0)
repairSwift1Harddrive = repair(132, "Swift 1", "Harddrive",
                                       56, 0.7, 0)
repairSwift1GPU = repair(133, "Swift 1", "GPU", 0, 1, 299)
repairSwift1CoolingSystem = repair(134, "Swift 1", "CoolingSystem",
                                       40, 0.7, 0)
repairSwift1Keyboard = repair(135, "Swift 1", "Keyboard",
                                       75, 0.8, 0)
repairSwift1Motherboard = repair(136, "Swift 1", "Motherboard",
                                       0, 1, 399)
repairSwift1Mousepad = repair(137, "Swift 1", "Mousepad",
                                       35, 0.8, 0)
repairSwift1Display = repair(138, "Swift 1", "Display", 0, 1, 199)
```

Table B- 15: Creating objects of 'repair' VI

```
repairAspire5CPU = repair(140, "Aspire 5", "CPU", 56, 0.7, 0)
repairAspire5Battery = repair(141, "Aspire 5", "Battery",
                                      75, 0.5, 0)
repairAspire5Harddrive = repair(142, "Aspire 5", "Harddrive",
                                      56, 0.7, 0)
repairAspire5GPU = repair(143, "Aspire 5", "GPU", 0, 1, 349)
repairAspire5CoolingSystem = repair(144, "Aspire 5",
                                      "CoolingSystem", 40, 0.7, 0)
repairAspire5Keyboard = repair(145, "Aspire 5", "Keyboard",
                                      63, 0.8, 0)
repairAspire5Motherboard = repair(146, "Aspire 5", "Motherboard",
                                      0, 1, 399)
repairAspire5Mousepad = repair(147, "Aspire 5", "Mousepad",
                                      35, 0.8, 0)
repairAspire5Display = repair(148, "Aspire 5", "Display", 0, 1, 199)
repairAspire7CPU = repair(150, "Aspire 7", "CPU", 89, 0.7, 0)
repairAspire7Battery = repair(151, "Aspire 7", "Battery",
                                      75, 0.5, 0)
repairAspire7Harddrive = repair(152, "Aspire 7", "Harddrive",
                                      56, 0.7, 0)
repairAspire7GPU = repair(153, "Aspire 7", "GPU", 0, 1, 479)
repairAspire7CoolingSystem = repair(154, "Aspire 7",
                                      "CoolingSystem", 40, 0.7, 0)
repairAspire7Keyboard = repair(155, "Aspire 7", "Keyboard",
                                      63, 0.8, 0)
repairAspire7Motherboard = repair(156, "Aspire 7", "Motherboard",
                                      0, 1, 479)
repairAspire7Mousepad = repair(157, "Aspire 7", "Mousepad",
                                      42, 0.8, 0)
repairAspire7Display = repair(158, "Aspire 7", "Display", 0, 1, 199)
```

Table B- 16: Creating a list containing all repair objects

```
listRepairs = [repairAspire5Battery, repairAspire5CoolingSystem,
               repairAspire5CPU, repairAspire5Display,
               repairAspire5GPU, repairAspire5Harddrive,
               repairAspire5Keyboard, repairAspire5Motherboard,
               repairAspire5Mousepad, repairAspire7Battery,
               repairAspire7CoolingSystem, repairAspire7CPU,
               repairAspire7Display, repairAspire7GPU,
               repairAspire7Harddrive, repairAspire7Keyboard,
               repairAspire7Motherboard, repairAspire7Mousepad,
               repairhpEliteBook830G6Battery,
               repairhpEliteBook830G6CoolingSystem,
               repairhpEliteBook830G6CPU,
               repairhpEliteBook830G6Display,
               repairhpEliteBook830G6GPU,
               repairhpEliteBook830G6Harddrive,
               repairhpEliteBook830G6Keyboard,
               repairhpEliteBook830G6Motherboard,
               repairhpEliteBook830G6Mousepad,
               repairhpEliteBook850G6Battery,
               repairhpEliteBook850G6CoolingSystem,
               repairhpEliteBook850G6CPU,
               repairhpEliteBook850G6Display,
               repairhpEliteBook850G6GPU,
               repairhpEliteBook850G6Harddrive,
               repairhpEliteBook850G6Keyboard,
               repairhpEliteBook850G6Motherboard,
               repairhpEliteBook850G6Mousepad,
               repairhpEliteBook1040G4Battery,
               repairhpEliteBook1040G4CoolingSystem,
               repairhpEliteBook1040G4CPU,
               repairhpEliteBook1040G4Display,
               repairhpEliteBook1040G4GPU,
               repairhpEliteBook1040G4Harddrive,
               repairhpEliteBook1040G4Keyboard,
               repairhpEliteBook1040G4Motherboard,
               repairhpEliteBook1040G4Mousepad,
               repairhpEliteBook1050G1Battery,
               repairhpEliteBook1050G1CoolingSystem,
               repairhpEliteBook1050G1CPU,
               repairhpEliteBook1050G1Display,
               repairhpEliteBook1050G1GPU,
               repairhpEliteBook1050G1Harddrive,
               repairhpEliteBook1050G1Keyboard,
               repairhpEliteBook1050G1Motherboard,
               repairhpEliteBook1050G1Mousepad,
               repairInspirion135000Battery,
               repairInspirion135000CoolingSystem,
               repairInspirion135000CPU,
               repairInspirion135000Display,
               repairInspirion135000GPU,
               repairInspirion135000Harddrive,
               repairInspirion135000Keyboard,
               repairInspirion135000Motherboard,
               repairInspirion135000Mousepad,
               repairMacBookAirBattery,
               repairMacBookAirCoolingSystem, repairMacBookAirCPU,
               repairMacBookAirDisplay, repairMacBookAirGPU,
               repairMacBookAirHarddrive, repairMacBookAirKeyboard,
```

```
                    repairMacBookAirMotherboard,repairMacBookAirMousepad,
                    repairMacBookProBattery,
                    repairMacBookProCoolingSystem, repairMacBookProCPU,
                    repairMacBookProDisplay, repairMacBookProGPU,
                    repairMacBookProHarddrive, repairMacBookProKeyboard,
                    repairMacBookProMotherboard,
                    repairMacBookProMousepad,
                    repairLifebookA357Battery,
                    repairLifebookA357CoolingSystem,
                    repairLifebookA357CPU, repairLifebookA357Display,
                    repairLifebookA357GPU, repairLifebookA357Harddrive,
                    repairLifebookA357Keyboard,
                    repairLifebookA357Motherboard,
                    repairLifebookA357Mousepad,
                    repairLifebookU728Battery,
                    repairLifebookU728CoolingSystem,
                    repairLifebookU728CPU,
                    repairLifebookU728Display,
                    repairLifebookU728GPU,
                    repairLifebookU728Harddrive,
                    repairLifebookU728Keyboard,
                    repairLifebookU728Motherboard,
                    repairLifebookU729Battery,
                    repairLifebookU729CoolingSystem,
                    repairLifebookU729CPU,
                    repairLifebookU729Display,
                    repairLifebookU729GPU, repairLifebookU729Harddrive,
                    repairLifebookU729Keyboard,
                    repairLifebookU729Motherboard,
                    repairLifebookU729Mousepad,
                    repairPrecision5520Battery,
                    repairPrecision5520CoolingSystem,
                    repairPrecision5520CPU, repairPrecision5520Display,
                    repairPrecision5520GPU, repairPrecision5520Harddrive,
                    repairPrecision5520Keyboard,
                    repairPrecision5520Motherboard,
                    repairPrecision5520Mousepad,repairSwift1Battery,
                    repairSwift1CoolingSystem, repairSwift1CPU,
                    repairSwift1Display, repairSwift1GPU,
                    repairSwift1Harddrive, repairSwift1Keyboard,
                    repairSwift1Motherboard, repairSwift1Mousepad,
                    repairThinkpadE490Battery,
                    repairThinkpadE490CoolingSystem,
                    repairThinkpadE490CPU,
                    repairThinkpadE490Display, repairThinkpadE490GPU,
                    repairThinkpadE490Harddrive,
                    repairThinkpadE490Keyboard,
                    repairThinkpadE490Motherboard,
                    repairThinkpadE490Mousepad,
                    repairThinkpadT580Battery,
                    repairThinkpadT580CoolingSystem,
                    repairThinkpadT580CPU, repairThinkpadT580Display,
                    repairThinkpadT580GPU, repairThinkpadT580Harddrive,
                    repairThinkpadT580Keyboard,
                    repairThinkpadT580Motherboard,
                    repairThinkpadT580Mousepad]
```

Table B- 17: Definition of method 'calculateRepairCosts'

```python
def calculateRepairCosts(laptop):
    sumcosts = 0
    defectComponents = ["", "", "", "", "", "", "", "", ""]
    counter = 0

    if (laptop.CPUCond == 1):
        defectComponents[counter] = "CPU"
        counter = counter + 1
    if (laptop.GPUCond == 1):
        defectComponents[counter] = "GPU"
        counter = counter + 1
    if (laptop.CoolingSystemCond == 1):
        defectComponents[counter] = "CoolingSystem"
        counter = counter + 1
    if (laptop.BatteryCond == 1):
        defectComponents[counter] = "Battery"
        counter = counter + 1
    if (laptop.DisplayCond == 1):
        defectComponents[counter] = "Display"
        counter = counter + 1
    if (laptop.KeyboardCond == 1):
        defectComponents[counter] = "Keyboard"
        counter = counter + 1
    if (laptop.MousepadCond == 1):
        defectComponents[counter] = "Mousepad"
        counter = counter + 1
    if (laptop.MotherboardCond == 1):
        defectComponents[counter] = "Motherboard"
        counter = counter + 1
    if (laptop.HarddriveCond == 1):
        defectComponents[counter] = "Harddrive"
        counter = counter + 1

    for j in range(0, len(defectComponents)):
        for i in range(0, len(listRepairs)):
            if ((laptop.Model == listRepairs[i].Model) &
               (defectComponents[j] == listRepairs[i].Component)):
                sumcosts = sumcosts + listRepairs[i].RepairCost

                i = i + 1

            else:
                i = i + 1

        j = j + 1

    return sumcosts

```

Table B- 18: Definition of method 'getSellingPrice'

```
def getSellingPrice(laptop):
    price = 0
    for i in range(0, len(listLaptopModels)):
        if (listLaptopModels[i].Model == laptop.Model):
            price = listLaptopModels[i].SellingPrice
            break
        else:
            i = i + 1
    return price

```

Table B- 19: Definition of method 'getValueResources'

```
def getValueResources(laptop):
    value = 0
    for i in range(0, len(listLaptopModels)):
        if (listLaptopModels[i].Model == laptop.Model):
            value = round((listLaptopModels[i].NewPrice) * 0.04, 0)
            break
        else:
            i = i + 1
    return value

```

Table B- 20: Definition of method 'calculateSavedEmissions'

```
def calculateSavedEmissions(usedLaptop):

    savedEmissions = 214

    if (usedLaptop.BatteryCond == 1):
        savedEmissions = savedEmissions - emissionsSpareBattery
    if (usedLaptop.CPUCond == 1):
        savedEmissions = savedEmissions - emissionsSpareCPU
    if (usedLaptop.GPUCond == 1):
        savedEmissions = savedEmissions - emissionsSpareGPU
    if (usedLaptop.DisplayCond == 1):
        savedEmissions = savedEmissions - emissionsSpareDisplay
    if (usedLaptop.KeyboardCond == 1):
        savedEmissions = savedEmissions - emissionsSpareKeyboard
    if (usedLaptop.MousepadCond == 1):
        savedEmissions = savedEmissions - emissionsSpareMousepad
    if (usedLaptop.MotherboardCond == 1):
        savedEmissions = savedEmissions - emissionsSpareMotherboard
    if (usedLaptop.OuterCasingCond == 1):
        savedEmissions = savedEmissions - emissionsSpareOuterCasing
    if (usedLaptop.HarddriveCond == 1):
        savedEmissions = savedEmissions - emissionsSpareHarddrive
    if (usedLaptop.CoolingSystemCond == 1):
        savedEmissions = savedEmissions -
               emissionsSpareCoolingSystem

    return (savedEmissions)
```

Table B- 21: Definition of method 'evaluateUsedLaptop'

```
def evaluateUsedLaptop(usedlaptop):
    sellingPriceLaptop = 0
    valueResources = 0
    valueResources = getValueResources(usedlaptop)
    savedEmissions = calculateSavedEmissions(usedlaptop)
    sellingPriceLaptop = getSellingPrice(usedlaptop)

    refurbishCosts = hourlyRateRefurbishing *
                               labourHoursRefurbishing

    repairCosts = calculateRepairCosts(usedlaptop)
    profit = sellingPriceLaptop * profitMargin
    riskAddOn = sellingPriceLaptop * riskProportion

    capitalCommitmentCosts = (sellingPriceLaptop - profit -
                               riskAddOn - repairCosts -
                               refurbishCosts)*(interestRate *
                               averageStorageTime)

    sumcosts = round(profit + riskAddOn + capitalCommitmentCosts +
                       repairCosts + refurbishCosts, 0)

    possiblePurchasePrice = sellingPriceLaptop - sumcosts

    if ((possiblePurchasePrice) <= 0):
        print("We suggest to recycle this laptop. We would like to
               purchase your laptop for ", valueResources, " €")
    else:
        print("We would like to purchase your Laptop. We are able to
               offer you ", possiblePurchasePrice, " €")
        print("You will save ", savedEmissions, " kgCo2eq by
               returning your laptop.")
```

Table B- 22: Definition of method 'predictBattery'

```
def predictBattery(potLaptop):
    data = {
        "Inputs": {
            "input1":
                [
                    {
                        'model': potLaptop.Model,
                        'age': potLaptop.Age,
                        'typeOfUse': potLaptop.TypeOfUse,
                        'mobileUse': potLaptop.MobileUse,
                        'useOfCover': potLaptop.UseOfCase,
                        'price': potLaptop.NewPrice,
                        'brand': potLaptop.Brand,
                        'coverOutside': potLaptop.TopCasingOutside,
                        'display': potLaptop.Display,
                        'baseOutside':potLaptop.BottomCasingOutside,
                        'baseInside': potLaptop.BottomCasingInside,
                        'keyboard': potLaptop.Keyboard,
                        'mousepad': potLaptop.Mousepad,
                        'conditionBattery': "0",
                        'conditionCPU': "0",
                        'conditionGPU': "0",
                        'conditionDisplay': "0",
                        'conditionKeyboard': "0",
                        'conditionMousePad': "0",
                        'conditionMotherboard': "0",
                        'conditionCover': "0",
                        'conditionHardDrive': "0",
                        'conditionCoolingSystem': "0",
                    }
                ],
        },
        "GlobalParameters": {
        }
    }
    body = str.encode(json.dumps(data))
    url = 'https://ussouthcentral.services.azureml.net/...
    api_key = 'P2W4l+Y4h/XI/rKOi8JzFszQAL ...
    headers = {'Content-Type': 'application/json',
               'Authorization': ('Bearer  ' + api_key)}
    req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    x = json.loads(result);
    r = x["Results"]['output1'][0]['Scored Labels']
    if r == "1":
        return (1)
    else:
        return (0)

except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

This function is executed in the same way for each of the ten components with a different url-adress and api_key.

Table B- 23: Definition of method 'createUsedLaptop'

```
def createUsedLaptop(potLaptop):

    usedLaptop = UsedLaptop(1, potLaptop.Model, potLaptop.Brand, "",
                            predictCPU(potLaptop),
                            predictGPU(potLaptop),
                            predictCoolingSystem(potLaptop),
                            predictBattery(potLaptop),
                            predictDisplay(potLaptop),
                            predictKeyboard(potLaptop),
                            predictMousepad(potLaptop),
                            predictMotherboard(potLaptop),
                            predictHousing(potLaptop),
                            predictHardDrive(potLaptop))
    return usedLaptop

```

Table B- 24: Creating GUI I

```
class Window(QWidget):
    model = ""
    age = 0
    type = ""
    mobile = 0
    bag = 0
    price = 0
    brand = ""
    cover = ""
    display = ""
    bottomOut = ""
    bottomIn = ""
    keyboard = ""
    mousepad = ""


    def __init__(self):
        super().__init__()
        self.initMe()

    def initMe(self):
        self.labelBrand = QLabel("Brand", self)
        self.labelBrand.move(80,45)
        self.cbBrand = QComboBox(self)
        self.cbBrand.addItems(["Select", "Acer", "Apple", "Dell",
                               "Fujitsu", "HP", "Lenovo"])
        self.cbBrand.move(80, 65)
        self.cbBrand.currentIndexChanged.connect(self.brand)
```

Table B- 25: Creating GUI II

```
self.cbBrand.move(80, 65)
        self.cbBrand.currentIndexChanged.connect(self.brand)

        self.labelModel = QLabel("Model", self)
        self.labelModel.move(80,110)
        self.cbModel = QComboBox(self)
        self.cbModel.addItems(["Select", "Fujitsu Lifebook A357",
                               "Fujitsu Lifebook U728", "Fujitsu
                               Lifebook U729", "MacBook Pro",
                               "MacBook Air", "HP EliteBook 830 G6",
                               "HP EliteBook 850 G6", "HP EliteBook
                               1040 G4", "HP EliteBook 1050 G1",
                               "Dell Inspirion 13 5000",  "Dell
                               Precision 5520", "Lenovo Thinkpad
                               T580", "Lenovo Thinkpad E490", "Acer
                               Swift 1", "Acer Aspire 5", "Acer
                               Aspire 7",])
        self.cbModel.move(80,130)
        self.cbModel.currentIndexChanged.connect(self.model)

        self.labelPrice = QLabel("New price", self)
        self.labelPrice.move(80,175)
        self.qlPrice = QLineEdit(self)
        self.qlPrice.move(80, 195)
        self.qlPrice.textChanged.connect(self.price)
        self.qlPrice.setValidator(QIntValidator())

        self.labelAge = QLabel("Age [months]", self)
        self.labelAge.move(80, 240)
        self.sbAge = QSpinBox(self)
        self.sbAge.move(80, 260)
        self.sbAge.valueChanged.connect(self.age)

        self.labelType = QLabel("Type of Use", self)
        self.labelType.move(80, 305)
        self.cbType = QComboBox(self)
        self.cbType.addItems(["Select", "surfing", "normal
                               business", "computationally
                               intensive", "gaming", "videos"])
        self.cbType.move(80, 325)
        self.cbType.currentIndexChanged.connect(self.type)

        self.labelMobile = QLabel("Laptop in mobile use", self)
        self.labelMobile.move(80, 370)
        self.cbMobile = QComboBox(self)
        self.cbMobile.addItems(["Select","No", "Yes"])
        self.cbMobile.move(80, 390)
        self.cbMobile.currentIndexChanged.connect(self.mobile)

        self.labelBag = QLabel("Use of laptop bag", self)
        self.labelBag.move(80, 435)
        self.cbBag = QComboBox(self)
        self.cbBag.addItems(["Select", "No", "Yes"])
        self.cbBag.move(80, 455)
        self.cbBag.currentIndexChanged.connect(self.bag)
```

Table B- 26: Creating GUI III

```
self.labelUpperCasing = QLabel("Condition top casing", self)
        self.labelUpperCasing.move(380, 45)
        self.cbUpperCasing = QComboBox(self)
        self.cbUpperCasing.addItems(["Select", "All right",
                             "Scratch", "Dent", "BrokenOutPart"])
        self.cbUpperCasing.move(380, 65)
        self.cbUpperCasing.currentIndexChanged.connect
                                (self.upperCasing)

        self.labelDisplay = QLabel("Condition display", self)
        self.labelDisplay.move(380, 110)
        self.cbDisplay = QComboBox(self)
        self.cbDisplay.addItems(["Select", "All right", "Defect",
                                "PixelErrors", "Scratch",
                                "BrokenScreen"])
        self.cbDisplay.move(380, 130)
        self.cbDisplay.currentIndexChanged.connect(self.display)

        self.labelBottomCasingOutside = QLabel("Condition bottom
                                casing  outside", self)
        self.labelBottomCasingOutside.move(380, 175)
        self.cbBottomCasingOutside = QComboBox(self)
        self.cbBottomCasingOutside.addItems(["Select", "All right",
                                        "Scratch", "Dent",
                                        "BrokenOutPart",])
        self.cbBottomCasingOutside.move(380, 195)
        self.cbBottomCasingOutside.currentIndexChanged.connect(self.
                                bottomCasingOutside)

        self.labelBottomCasingInside = QLabel("Condition bottom
                                        casing inside", self)
        self.labelBottomCasingInside.move(380, 240)
        self.cbBottomCasingInside = QComboBox(self)
        self.cbBottomCasingInside.addItems(["Select", "All right",
                                        "Scratch", "Dent",
                                        "BrokenOutPart",])
        self.cbBottomCasingInside.move(380, 260)
        self.cbBottomCasingInside.currentIndexChanged.connect
                                self.bottomCasingInside)

        self.labelKeyboard = QLabel("Condition keyboard", self)
        self.labelKeyboard.move(380, 305)
        self.cbKeyboard = QComboBox(self)
        self.cbKeyboard.addItems(["Select", "All right", "Highly
                                contaminated",
                                "Missing key", "Defect",])
        self.cbKeyboard.move(380, 325)
        self.cbKeyboard.currentIndexChanged.connect(self.keyboard)
```

Table B- 27: Creating GUI IV

```
            self.labelMousepad = QLabel("Condition mousepad", self)
            self.labelMousepad.move(380, 370)
            self.cbMousepad = QComboBox(self)
            self.cbMousepad.addItems(["Select", "All right", "Scratch",
                                       "Defect",])
            self.cbMousepad.move(380, 390)
            self.cbMousepad.currentIndexChanged.connect(self.mousepad)

            button = QPushButton('Submit', self)
            button.move(600,250)
            button.clicked.connect(self.submit)
            self.setGeometry(50, 50, 700, 550)
            self.setWindowTitle("Laptop evaluation")
            self.show()

            def model(self, i):
                     w.model = self.cbModel.currentText()

            def brand(self, i):
                    w.brand = self.cbBrand.currentText()

            def age(self):
                    w.age = self.sbAge.value()

    def type(self):
        w.type = self.cbType.currentText()

    def mobile(self):
        if(self.cbMobile.currentText() == "Yes"):
            w.mobile = 1
        else:
            if (self.cbMobile.currentText() == "No"):
               w.mobile = 0

          def bag(self):
            if(self.cbBag.currentText() == "Yes"):
                w.bag = 1
            else:
                if (self.cbBag.currentText() == "No"):
                    w.bag = 0

        def price(self, text):
            w.price = self.qlPrice.text()

        def upperCasing(self):
            w.cover = self.cbUpperCasing.currentText()

        def display(self):
            w.display = self.cbDisplay.currentText()

        def bottomCasingOutside(self):
            w.bottomOut = self.cbBottomCasingOutside.currentText()
```

Table B- 28: Creating GUI V

```
    def bottomCasingInside(self):
        w.bottomIn = self.cbBottomCasingInside.currentText()

    def keyboard(self):
        w.keyboard = self.cbKeyboard.currentText()

    def mousepad(self):
        w.mousepad = self.cbMousepad.currentText()

    def submit(self):
        global z

        z.Model = w.model
        z.Age = w.age
        z.TypeOfUse = w.type
        z.MobileUse = w.mobile
        z.UseOfCase = w.bag
        z.NewPrice = w.price
        z.Brand = w.brand
        z.TopCasingOutside = w.cover
        z.Display = w.display
        z.BottomCasingOutside = w.bottomOut
        z.BottomCasingInside = w.bottomIn
        z.Keyboard = w.keyboard
        z.Mousepad = w.mousepad

        evaluateUsedLaptop(createUsedLaptop(z))

app = QApplication(sys.argv)
w = Window()

sys.exit(app.exec_())
```

Appendix C Surveys

Appendix C.1 **Information letter experts**

Dear Ms/Mr Example,

My name is Michael Diem and I study Digital Industrial Management and Engineering at Stellenbosch University and Reutlingen University.

Would you be interested to answer a short survey regarding the decision-making process of End-of-Usage products in the Circular Economy?

The developed decision-making process determines the next step of an End-of-Usage product in the Circular Economy. To do this, the laptop owner must enter easily identifiable information about his/her laptop via a graphical user interface. Based on this information, artificial neural networks determine the conditions of the laptop components (whether they need to be replaced because they are defective or have a short remaining lifetime, or can remain in the laptop). This information is used to determine whether the laptop can be reused directly, new components need to be installed or the laptop can only be recycled. This is done by calculating the costs that would incurred for the different recovery options. Repair costs are calculated for the components, which need to be replaced. These costs will be deducted from the possible sales price. If the difference is positive, it will be offered to the owner of the laptop as a price. In addition, the number of emissions saved by recycling the laptop (based on the component conditions) is communicated to the owner. If the difference is negative, the owner is offered a price for the resources contained in the laptop.

The laptop does not have to be sent in for this process. The owner of the laptop can do it from home via a homepage or app. The following information is required: brand; model; age; mobile use (yes/no); use of protective case (yes/no); type of use (5 different classes); predefined damages to 6 parts of the laptop (e.g. dent in the bottom casing).

A comprehensive amount of training data is required to train the artificial neural networks. This means data sets which contain the information of the owner as well as component conditions determined by an expert. So far only simulated data were used, which led to good results. The next step will be to work with real data.

Please find attached two videos, which demonstrate a process by the user. In the first run, a laptop in good condition is demonstrated, which can be sold again with little effort. In the second run an older laptop in bad condition is demonstrated, which can only be recycled.

The following link will take you to the survey:

https://docs.google.com/forms/d/e/1FAIpQLScQ-HSicQsjc4J5lIookofQY32F9uWONbQo6CbkGgHN2uhTDA/viewform?usp=sf_link

Thank you in advance for your time.

Kind regards

Michael Diem

Appendix C.2 **Questions experts**

What is your initial reaction regarding the economic and ecological relevance of the topic? *

- ◯ Very relevant
- ◯ Relevant
- ◯ Neutral
- ◯ Less relevant
- ◯ Not relevant

Figure C- 1: 1. Question experts

How do you assess the feasibility of intergrating/ implementation of the developed process into existing processes? *

- ◯ Easy feasible
- ◯ Feasible
- ◯ Difficult to implement
- ◯ Not feasible
- ◯ Not enough information

Figure C- 2: 2. Question experts

Do you agree with the selected machine learning method (artificial neural network) to determine the condition of the laptop components? *

- Strongly agree
- Agree
- Disagree
- Strongly disagree
- Not enough information

Figure C- 3: 3. Question experts

Do you agree with the selected input information (laptop information, stress factors and external damages) in order to create a decentralised decision? *

- Strongly agree
- Agree
- Disagree
- Strongly disagree
- Not enough information

Figure C- 4: 4. Question experts

How do you assess the economic benefit of the developed process? *

- Very positive
- Positive
- Negative
- Very negative
- Not enough information

Figure C- 5: 5. Question experts

How do you assess the ecological benefit of the developed process (prevention of emissions and increase of resource efficiency)? *

- ◯ Very positive
- ◯ Positive
- ◯ Negative
- ◯ Very negative
- ◯ Not enough information

Figure C- 6: 6. Question experts

How do you rate the ability of the process to deal with other electronic products (after adaptation of variables)? *

- ◯ Very High
- ◯ High
- ◯ Low
- ◯ Very low
- ◯ Not enough information

Figure C- 7: 7. Question experts

How do you rate the ability of the process to deal with other NON-electronic products (after adaptation of variables)? *

- Very High
- High
- Low
- Very low
- Not enough information

Figure C- 8: 8. Question experts

How do you rate the attractiveness of the developed process for the owner of an End-of-Usage product regarding the incentive system, simplicity and ecological information? *

- Very High
- High
- Low
- Very low
- Not enough information

Figure C- 9: 9. Question experts

Please indicate your business sector and job profile. The information will only be used to classify the participants of the survey. *

Figure C- 10: 10. Question experts

Appendix C.3 **Information letter users**

Dear Ms/Mr Example,

My name is Michael Diem and I study Digital Industrial Management and Engineering at Stellenbosch University and Reutlingen University.

Would you be interested to answer a short survey regarding the decision-making process of End-of-Usage products in the Circular Economy?

The developed decision-making process determines the next step of an End-of-Usage product in the Circular Economy. If a laptop owner decides not to use her/his laptop anymore, she/he can go through this process. Information about the laptop is required. These are brand; model; age; mobile use (yes/no); use of protective case (yes/no); type of use (5 different classes); predefined damages to 6 parts of the laptop (e.g. dent in the bottom casing).

This information is used to determine whether the laptop can be reused directly, new components need to be installed or the laptop can only be recycled. This is done by calculating the costs that would incurred for the different recovery options. These costs will be deducted from the possible sales price. If the difference is positive, it will be offered to the owner of the laptop as a price. In addition, the number of emissions saved by recycling the laptop is communicated to the owner. If the difference is negative, the owner is offered a price for the resources contained in the laptop. The laptop does not have to be sent in for this process. The owner of the laptop can do it from home via a homepage or app.

Please find attached two videos, which show a process run by an user.

The following link will take you to the survey:

https://docs.google.com/forms/d/e/1FAIpQLSfg387FnhZYzhA3sV7bMpokrfK0UPAY33J2-QOfnN1RtlJHmg/viewform?usp=sf_link

Thank you in advance for your time.

Kind regards

Michael Diem

Appendix C.4 **Questions user**

What is your initial reaction regarding the developed decision-making process? *

- Very positive
- Positive
- Neutral
- Negative
- Very negative

Figure C- 11: 1. Question users

Would you use the developed decision-making process? *

- Definitely yes
- Probably
- Probably not
- Definitely not

Figure C- 12: 2. Question users

How easily can you determine the required information? *

- Very easy
- Easy
- Difficult
- Very difficult

Figure C- 13: 3. Question users

How attractive do you rate the decentralized execution of the process? *

- Very attractive
- Attractive
- Neutral
- Unattractive
- Very unattractive

Figure C- 14: 4. Question users

How attractive do you rate the information regarding the ecological impact of the decision? *

- Very attractive
- Attractive
- Neutral
- Unattractive
- Very unattractive

Figure C- 15: 5. Question users

How do your rate the simplicity of the execution of the developed process? *

- Very easy
- Easy
- Difficult
- Very difficult

Figure C- 16: 6. Question users